# Advantage™ VISION:Builder®

# Advantage™ VISION:Two™

## Reference Guide

### 14.0

Computer Associates™

BUUSR140.PDF/D92-001-014

This documentation and related computer software program (hereinafter referred to as the "Documentation") is for the end user's informational purposes only and is subject to change or withdrawal by Computer Associates International, Inc. ("CA") at any time.

This documentation may not be copied, transferred, reproduced, disclosed or duplicated, in whole or in part, without the prior written consent of CA. This documentation is proprietary information of CA and protected by the copyright laws of the United States and international treaties.

Notwithstanding the foregoing, the user may print a reasonable number of copies of this documentation for its own internal use, provided that all CA copyright notices and legends are affixed to each reproduced copy. Only authorized employees, consultants, or agents of the user who are bound by the confidentiality provisions of the license for the software of the user will have access to such copies.

This right to print copies is limited to the period during which the license for the product remains in full force and effect. Should the license terminate for any reason, it shall be the user's responsibility to return to CA the reproduced copies or to certify to CA that same have been destroyed.

To the extent permitted by applicable law, CA provides this documentation "as is" without warranty of any kind, including without limitation, any implied warranties of merchantability, fitness for a particular purpose or noninfringement. In no event will CA be liable to the end user or any third party for any loss or damage, direct or indirect, from the use of this documentation, including without limitation, lost profits, business interruption, goodwill, or lost data, even if CA is expressly advised of such loss or damage.

The use of any product referenced in this documentation and this documentation is governed by the end user's applicable license agreement.

The manufacturer of this documentation is Computer Associates International, Inc.

Provided with "Restricted Rights" as set forth in 48 C.F.R. Section 12.212, 48 C.F.R. Sections 52.227-19(c)(1) and (2) or DFARS Section 252.227-7013(c)(1)(ii) or applicable successor provisions.

© 2002 Computer Associates International, Inc. (CA).

All trademarks, trade names, service marks, and logos referenced herein belong to their respective companies.

# Contents

## Chapter 1: Introduction


Advanced Syntax Language ..... 1-1
About This Book ..... 1-2
  Structure ..... 1-2
  Terminology and Documentation Conventions ..... 1-4
Understanding VISION:Builder ..... 1-5
Contacting Computer Associates ..... 1-6


## Chapter 2: Application Development System


VISION:Builder Engine ..... 2-1
VISION:Workbench for DOS ..... 2-2
VISION:Workbench for ISPF ..... 2-3
VISION:Builder Library Utilities ..... 2-4
Discipline of VISION:Builder ..... 2-4
  Automatic Functions ..... 2-8
Using VISION:Builder ..... 2-10
  The Definition/ Maintenance Run ..... 2-10
  The Application Run ..... 2-12
Input and Output Files ..... 2-14
  Input Files ..... 2-14
  Output Files ..... 2-16
Transaction Processing❹ ..... 2-18
Tables ..... 2-20
Arrays ..... 2-20
Reporting ..... 2-21

# Chapter 3: Defining Data


Concept of Structured Files ... 3-2
- Important Points to Note About Hierarchical Structures ... 3-5
- Terms of Hierarchical Structure ... 3-5
- Defining a Structured File to VISION:Builder ... 3-6
- File Glossary ... 3-8

Entering a File Definition ... 3-9
- Catalog the File Definition in the Common Library ... 3-20

Defining a VSAM File ... 3-20
- VSAM Record Structures ... 3-21
- VSAM Cluster Definitions ... 3-21
- VSAM Alternate Indexing ... 3-22

Defining an IMS File ... 3-22
- IMS Record Structure ... 3-23
- VISION:Builder and IMS ... 3-23
- Write a DBD/PCB from a VISION:Builder File Definition ... 3-25
- Write a VISION:Builder File Definition from a DBD/PCB ... 3-29
- Secondary Indexing ... 3-31
- VISION:Builder File Definitions and Secondary Indexing ... 3-33
- Preparing a Secondary Structure Definition ... 3-40

Defining a Relational File ... 3-42
- Understanding Logical Records ... 3-42
- Advantages of Logical Records ... 3-43
- Defining the Logical Record ... 3-46
- The Relational File Definition ... 3-48
- Defining the Primary Key Field ... 3-86
- Defining the Match Field ... 3-88
- Constructing the Supplemental WH Statements ... 3-88
- Example of a Multi-Column Keyed Segment ... 3-88

GDBI Files ... 3-90

Compressed Files ... 3-90

Working Storage ... 3-93

Sample Definition/Maintenance Run ... 3-93

## Chapter 4: Field Characteristics and Handling


Character String Fields - Type C ..... 4-2
Zoned Decimal Number Fields - Type Z ..... 4-3
Fixed Point Number Fields - Type F ..... 4-3
Floating Point Number Fields - Type E ..... 4-3
Time Fields - Type L, S, or U ..... 4-3
Date Fields - Type D ..... 4-3
Packed Decimal Number Fields - Type P ..... 4-3
Extended Packed Decimal Fields ..... 4-4
Variable Length Fields (Type V) ..... 4-4
Variable Length Fields in Reports ..... 4-6
Restrictions on Using Variable Length Fields ..... 4-8
Time Fields (Types L, S, or P) ..... 4-8
Time Data Conversions and Arithmetic Operations ..... 4-9
Date Fields (Type D) ..... 4-10
Validation and Automatic Conversion ..... 4-10
Rules for Conversion in Arithmetic and Comparison Operations ..... 4-11
Rules for Conversion in Replacement Operations ..... 4-12
Output Edit Length ..... 4-14
Variable Length Field ..... 4-14
Character Field ..... 4-15
Packed, Zoned, and Fixed Point Fields ..... 4-15
Floating Point ..... 4-16
Date Field ..... 4-16
Temporary Fields ..... 4-16
Implicitly Defined Temporary Fields ..... 4-17
Explicitly Defined Temporary Fields ..... 4-17
Invalid, Missing, and Overflow Fields ..... 4-18
Invalid Fields ..... 4-18
Missing Fields (Non-Existent Fields) ..... 4-19
Field Overflow ..... 4-19
Partial Fielding a Character Field ..... 4-20

## Chapter 5: The Application Cycle


The Common Library in Basic Processing ..... 5-2
Application Cycle Overview ..... 5-3
Basic Processing Work Areas ..... 5-5
- Work Areas, Procedures, Reports, and Subfiles ..... 5-5
- Flag Fields ..... 5-6

Application Components ..... 5-7
- Procedures ..... 5-8
- Reports ..... 5-9
- Subfiles ..... 5-10

Basic Application Cycle ..... 5-11
Application Cycle and Multiple Reports ..... 5-14
Sort Step ..... 5-16
Report Step ..... 5-16
Single-Step Processing ..... 5-17
Sample Application Source Listing ..... 5-18


## Chapter 6: Input File Processing


The Master File ..... 6-4
- Master File Processing Options ..... 6-5

The Transaction File❹ ..... 6-9
Coordinated Files ..... 6-10
- Indexed Coordination ..... 6-11
- Sequential Coordination ..... 6-17
- User Coordination ..... 6-46

Source Statements ..... 6-59


## Chapter 7: Output File Processing


Master File ..... 7-3
Report Files ..... 7-4
Subfiles ..... 7-6
- Multiple Outputs to a Subfile ..... 7-10
- Using a Subfile to Update a Sequentially Coordinated File ..... 7-10
- Subfile Output to Direct-Access Data Sets ..... 7-10
- Subfile Output and IMS Databases ..... 7-10
- Subfile Output and Relational Tables ..... 7-11
- Subfile Output and Removing Dynamically Deleted Master File Segments❹ ..... 7-13

Report Summary Files ..... 7-14
- Record Format ..... 7-16
- Generated File Definition ..... 7-18

Rejected Transactions File❹ ..... 7-22
Deleted Master File Records❹ ..... 7-22
OS/390 System-Determined Block Size Support ..... 7-22
Source and Report Listings File ..... 7-23
Alternate List File for Report Listings ..... 7-24
Alternate Report Output Methods ..... 7-25
Comma-Delimited Data (CSV) ..... 7-26
Tab-Delimited Data ..... 7-26
Plain Text File ..... 7-26
HTML Document ..... 7-27
Raw Data Output ..... 7-27

## Chapter 8: Data Structure Handling

Automatic Looping ..... 8-1
Single-Path Nested Loops ..... 8-5
Multi-Path Nested Loops ..... 8-8
Multi-Path Parallel Looping ..... 8-11
Examples of Single-Path Nested Loops ..... 8-12
Examples of Multi-Path Looping ..... 8-17
Selection Control Looping ..... 8-20
Segment Operators in Looping ..... 8-21
Subroutine Looping ..... 8-24
Branching to Control Looping ..... 8-26

## Chapter 9: Reporting

Definition of the Inventory File ..... 9-1
Generating Simple Reports ..... 9-2
Sequencing and Grouping Report Records ..... 9-8
Controlling the Sequence of Report Records ..... 9-8
Grouping Report Data ..... 9-8
Levels of Control Break ..... 9-9
Summarizing Report Data ..... 9-11
Creating Subtitles and Adjusting Summary Labels ..... 9-13
Reports Using Summaries Only ..... 9-14
Field Output Edit ..... 9-16
Reporting Optimization ..... 9-19
Report Page Layout ..... 9-20

## Chapter 10: Transactions

Transaction File and Master File Record Flows ..... 10-4
Transaction File and Sequential Master File Record Flows ..... 10-5
Transaction File and Direct Master File Record Flows ..... 10-6
Transaction Processing Applications ..... 10-7
Transaction Definitions ..... 10-7

## Chapter 11: Requests

Request Sections ..... 11-1
Request Types ..... 11-4
Request Sequencing ..... 11-4
Request Branching ..... 11-5
Branch Locations ..... 11-7
Subroutine Requests and Subroutine Sets ..... 11-11
Subroutine Request ..... 11-11
Subroutine Request Sets ..... 11-13
Back Branching ..... 11-13
Requests and ASL Procedures ..... 11-15

## Chapter 12: Tables

Table Types ..... 12-2
Binary Table ..... 12-2
Sequential Table ..... 12-3
Displacement Table ..... 12-4
Defining Tables ..... 12-5
Table Lookup Functions ..... 12-6
Automatic Table Lookup ..... 12-6
Procedural Table Lookup ..... 12-7

## Chapter 13: Arrays


Array Definitions ..... 13-3
Array Looping ..... 13-5
- Simple Array Looping ..... 13-5
- Nested Array Looping ..... 13-6
- Parallel Array Looping ..... 13-7
- Controlling Array Looping ..... 13-9
- Array and Automatic Looping ..... 13-13

Array Processing ..... 13-13
- Loading an Array in Memory ..... 13-14
- Reporting from Arrays ..... 13-21
- Array Equivalence - Overdefining Arrays ..... 13-23
- Output an Array as a Subfile ..... 13-27
- Input an Array as a Coordinated File ..... 13-27


## Chapter 14: Extended Reporting Capabilities


Formatted Sectional Reporting ..... 14-1
- Formatted Sectional Reporting - Syntax ..... 14-2
- Using Formatted Sectional Reporting ..... 14-5

Formatted Reporting ..... 14-13
- Formatted Reporting - Syntax ..... 14-13
- Using Formatted Reporting ..... 14-16

Variable Length Fields ..... 14-18

When to Use Formatted Sectional Reporting ..... 14-19

Dynamic Report Line Modification ..... 14-19
- Processing ..... 14-20
- Errors ..... 14-23
- Syntax ..... 14-23
- Storage Considerations ..... 14-23
- Restrictions and Operating Characteristics ..... 14-24
- Using Dynamic Report Line Modification ..... 14-24
- Dynamic Report Line Modification with Multi-Image Reports ..... 14-25

Graphics ..... 14-27
- Types of Graphs ..... 14-27
- Producing Scatter Diagrams ..... 14-32
- Producing Bar Graphs ..... 14-38
- Technical Considerations and Restrictions ..... 14-44

Painting a Report Section in VISION:Workbench for DOS ..... 14-49

## Chapter 15: Advanced Transaction Processing


Transaction Processing Step ..... 15-3
Transaction Procedures ..... 15-6
- Type 1 Procedure Processing / Transaction Record Identification/Validation ..... 15-12
- Alignment and Type 2 Procedure Processing ..... 15-13
- Updating and Type M and Type 3 Procedure Processing Flow ..... 15-15
- Transaction Record Rejection and Type 4 Procedure Processing Flow ..... 15-16

Using Transaction Procedures for an Updating Application ..... 15-17
- Elements of the Updating Run ..... 15-19

The Update Application Flow ..... 15-23
The Update Application Reports and Outputs ..... 15-34
Transaction and Master Files for Applications Other than Updating ..... 15-39
IMS Updating Considerations ..... 15-42
Relational Updating Considerations ..... 15-42
- Set Operations ..... 15-45
- Referential Integrity ..... 15-49
- Designing for Referential Integrity ..... 15-51
- The Order of Updating Rows ..... 15-56


## Chapter 16: Database Access Methodology


IMS Processing ..... 16-1
- Standard Processing ..... 16-3
- Memory Optimized (MOSAIC) Processing ..... 16-5
- Preselection ..... 16-9
- VISION:Builder Processing and Secondary Indexing ..... 16-10
- Storage Utilization ..... 16-11
- Technical Considerations and Hints ..... 16-11

Relational Support for DB2 ..... 16-12
- Setting Up and Running a Job for DB2 ..... 16-12
- How VISION:Builder Constructs DB2 Statements ..... 16-13
- Uniqueness of Rows in Relational Tables ..... 16-16
- Foreign Keys ..... 16-16
- Validation Against the DB2 Catalog ..... 16-16
- COMMITs During VISION:Builder Processing ..... 16-17
- Performance Considerations ..... 16-17
- NULL Columns ..... 16-18
- Multi-Column Keyed Segments ..... 16-18

## Chapter 17: Report Manager

Reporting Concepts ..... 17-3
Report Collating Overview ..... 17-5
Report Routing Overview ..... 17-7
Collating with Routing Overview ..... 17-9
Report Manager Specifications ..... 17-10
Considerations of Collating and Routing ..... 17-10
JCL Considerations ..... 17-12
Multiple Report Files ..... 17-12
Storage Considerations ..... 17-13
COLLATE and ROUTE Application Examples ..... 17-13

## Chapter 18: External Program Linkage

Linking to VISION:Builder from an External Program ..... 18-1
Linkage Section Conventions ..... 18-1
Defining the Linkage Section ..... 18-5
Calls from VISION:Builder to an External Program ..... 18-8
Specifying External Calls ..... 18-8
Calls to IMS ..... 18-22
Calls to DL/I VSE Environment ..... 18-23
Calls to IBM Language Environment Services ..... 18-24

## Chapter 19: Cataloging

Definitions ..... 19-1
Instream File Definitions ..... 19-2
Run Data Groups ..... 19-4
Procedures, Requests, and Request Groups ..... 19-5
Item Tracking ..... 19-6
Processing with Multiple Common Libraries ..... 19-7

## Chapter 20: Program Analyzer


Numeric Conversion Messages ..... 20-2
Generating the M4PAOUT File ..... 20-4
Processing the M4PAOUT File ..... 20-6
Cross Reference Reports ..... 20-8
- Field Name Cross Reference Report ..... 20-8
- Constant/Pattern Cross Reference Report ..... 20-12
- Branch Location Cross Reference Report ..... 20-14
- File Cross Reference Report ..... 20-15

Request Execution Trace Reports ..... 20-17
- The Request Execution Trace Report ..... 20-17
- The GDBI Mapping Request Execution Trace Report ..... 20-20

Using PAL to Help Maintain a Program ..... 20-21
- What Fields are Being Used? ..... 20-22
- What Files are Used or Created? ..... 20-23

Source that Generates Reports ..... 20-24
- File Definitions ..... 20-24
- The Requests ..... 20-27
- Creating the Field Cross Reference Report ..... 20-30
- Creating the Constant/Pattern Cross Reference Report ..... 20-32
- Creating the Branch Location Cross Reference Report ..... 20-33
- Creating the File Name Cross Reference Report ..... 20-34
- Creating the Execution Trace Reports ..... 20-36


## Appendix A: Sample DB2 Applications

## Appendix B: VISION:Builder System Component Relationships

## Appendix C: DB2 Reserved Words

## Appendix D: Complete Sample Applications


The Application Cycle ..... D-1
Input File Processing ..... D-5
- Preparation Job ..... D-5
- Examples Jobs ..... D-31

Output File Processing ..... D-60
Data Structure Handling ..... D-65
Reporting ..... D-77

Arrays ..... D-90
Preparation Job ..... D-90
Examples Jobs ..... D-98
Extended Reporting Capabilities ..... D-127
Formatted Sectional Report ..... D-127
Advanced Transaction Processing ..... D-137
Definition Run ..... D-137
ITEM Master Create Run ..... D-155
CUSTMAST Create Run ..... D-158
Updating Application ..... D-160
Header Trailer Application ..... D-176
Report Manager ..... D-179
MSTR File Definition Glossary ..... D-179
Standard Reports Application ..... D-180
COLLATE Reports Application ..... D-186
ROUTE Reports Application ..... D-193
COLLATE and ROUTE Reports Application ..... D-202
Cataloging ..... D-211

## Appendix E: Glossary

A ..... E-1
B ..... E-2
C ..... E-4
D ..... E-6
E ..... E-7
F ..... E-8
G-H ..... E-10
I-K ..... E-11
L ..... E-12
M ..... E-14
N-O ..... E-15
P-Q ..... E-16
R ..... E-17
S ..... E-20
T-U ..... E-24
V-W ..... E-27

## Index

Chapter

# 1 Introduction

This manual describes the important concepts in VISION:Builder® and follows these with illustrative examples wherever necessary. It is structured in chapters that build on each other. The chapters are relatively independent of each other once you have understood how VISION:Builder works. The chapters center around specific concepts in VISION:Builder such as reporting, definitions, processing, and updating.

This manual describes all of the concepts necessary to understand how VISION:Builder works and how applications can be developed using VISION:Builder. It attempts to be as complete as possible in covering all of the important functions of VISION:Builder. Throughout this manual, examples are presented to illustrate the concepts being described.

## Advanced Syntax Language

Most of the examples are coded using the Advanced Syntax Language (ASL), which is a freeform language used to build procedures consisting of logical, arithmetic, and data manipulation statements. For a detailed description, see the *ASL Reference Guide*. In some cases, the fixed format syntax statements are used for illustrating a concept. See the *VISION:Builder Reference Summary* for details on the fixed format syntax statements.

ASL is directly supported in its native freeform syntax when using the following application development tools:

- VISION:Workbench for DOS, in z/OS™, OS/390®, CMS, and VSE environments
- VISION:Workbench for ISPF, only in the OS/390 environment

See Chapter 2, *Application Development System* and the *VISION:Workbench for ISPF Reference Guide* for a description of the VISION:Workbench for ISPF component. When using the VISION:Builder Engine, ASL is directly supported in its native freeform syntax in the z/OS, OS/390, and CMS environments. Native ASL syntax is not directly supported in the VSE environment.

For VISION:Builder users in the VSE environment, VISION:Workbench for DOS will translate your native ASL syntax coding into fixed format statements during the export process. For details on specifying target host support of native ASL, refer to the Workbench Setup window description in the *VISION:Workbench for DOS Reference Guide*.

# About This Book

## Structure

This manual covers the fundamental principles of VISION:Builder.

| | |
|---|---|
| Chapter 1, *Introduction* | Describes this book and the features of VISION:Builder. |
| Chapter 2, *Application Development System* | Describes and illustrates the concepts and philosophy of VISION:Builder. |
| Chapter 3, *Defining Data* | Discusses the concept of structured files and how to define relational files, VSAM files, and IMS™ files. |
| Chapter 4, *Field Characteristics and Handling* | Discusses how VISION:Builder handles field conversions during various operations. Variable length and time fields are described in addition to character and numeric field types. The notion of invalid fields and missing fields is a key concept in this section. |
| Chapter 5, *The Application Cycle* | Describes the processing cycle inherent in a VISION:Builder application that does not use transaction processing. This chapter is an important conceptual foundation for understanding exactly how VISION:Builder works. |
| Chapter 6, *Input File Processing* | Discusses input files, including coordinated files. |
| Chapter 7, *Output File Processing* | Discusses output files, including subfiles. |
| Chapter 8, *Data Structure Handling* | Discusses automatic looping and controlled looping. |
| Chapter 9, *Reporting* | Covers simple reporting: the defaults VISION:Builder lets you use, the concept of control breaks, and automatic features (for example, totals, averages, and counts). |
| Chapter 10, *Transactions* | Discusses transaction definitions and using transactions in the transaction cycle. |

| | |
|---|---|
| Chapter 11, *Requests* | Discusses VISION:Builder requests and branching and subroutine concepts as they relate to requests. |
| Chapter 12, *Tables* | Details table definitions, usage, and benefits. |
| Chapter 13, *Arrays* | Discusses the concept of arrays. Shows how to define and use arrays. |
| Chapter 14, *Extended Reporting Capabilities* | Shows how reports can be formatted independently of VISION:Builder automatic reporting. Discusses the dynamic capability of manipulating a report line just prior to output. Illustrates producing graph reports using the GRAPHICS option. |
| Chapter 15, *Advanced Transaction Processing* | Discusses advanced transaction concepts and techniques. Shows how to customize the transaction cycle with procedures at various points in the cycle. |
| Chapter 16, *Database Access Methodology* | Discusses database considerations and performance issues relevant to different database access methods. |
| Chapter 17, *Report Manager* | Discusses the collating and routing capabilities of the Report Manager function. |
| Chapter 18, *External Program Linkage* | Discusses external program linkage. |
| Chapter 19, *Cataloging* | Discusses cataloging library items. |
| Chapter 20, *Program Analyzer* | Discusses the debugging and documentation capabilities of the VISION:Builder Program Analyzer. Shows examples of the cross reference and trace reports that the Program Analyzer can produce. |
| Appendix A, *Sample DB2 Applications* | Introduces a complete VISION:Builder application. Its objective is to print a glossary of table of contents for one or more DB2® tables. |
| Appendix B, *VISION:Builder System Component Relationships* | Describes the relationship among the components of VISION:Builder. In particular, it focuses on the relationship of the available user interface components, VISION:Workbench for DOS and VISION:Workbench for ISPF. |
| Appendix C, *DB2 Reserved Words* | Contains a list of DB2 reserved words. |

| | |
|---|---|
| Appendix D, *Complete Sample Applications* | Contains versions of sample applications that are used throughout this guide. |
| Appendix E, *Glossary* | Defines some of the commonly used words and phrases in VISION:Builder. |

## Terminology and Documentation Conventions

- If you are an IMS or DL/I user, whenever the term IMS is used in this manual, assume the term applies to both IMS and DL/I.
- If you are a VISION:Two™ user, whenever the term VISION:Builder is used in the documentation, assume the term VISION:Two. However, some features and options are not common to both VISION:Builder and VISION:Two. Where a feature or option does not apply, the symbol ❹ is used to designate that the feature or function being described only applies to the VISION:Builder 4000 model series and not to VISION:Two.
- Bolded words in code examples are particularly germaine to the discussion in the text.
- Throughout the documentation, in the report samples, a vertical ellipsis is used to specify that portions of the report sample have been removed, as shown in the following sample:

```
-----------------------------------------------------------------------------------------------------------------------------------
   RUN     STMT    FILE    O N S D U T R A B S D S L  RB  R S M R S            START            END              SEQUENCE   LIST
   NAME    TYPE    NAME    L E E I P R E U U R L C S  PL  J O O F S            SEARCH          SEARCH            73 TO 80    SEQ
                           D W Q R D N P D F T M N T  TK  T P P O R
-----------------------------------------------------------------------------------------------------------------------------------
(USERCORD) (RC) (CUSTOMER) (S)          (S)                     (R)                                                (00880000)     1
-----------------------------------------------------------------------------------------------------------------------------------
   RUN     STMT  LOGICAL  LBL  DTF/DD      KEY       KEY       KEY      QUALIFIER   USER I/O  ARRAY       PASSWORD/  SEQUENCE   LIST
   NAME    TYPE  FILENAME       NAME     FIELD-1   FIELD-2   FIELD-3   ICF CRD DBI   MODULE   QUAL        AUTH.-ID   73 TO 80    SEQ
-----------------------------------------------------------------------------------------------------------------------------------
(USERCORD) (RF) (ORDERS  ) (S) (M4CORD1 )                                (R)                                        (00890000)     2
(USERCORD) (RF) (ORDERS  ) (S) (M4CORD2 )                                (R)                                        (00890000)     3
(USERCORD) (RF) (ORDERS  ) (S) (M4CORD3 ) (CUSTNUM )                     (G)                                        (00890000)     4
(USERCORD) (RF) (ORDERS  ) (S) (M4CORD4 ) (CUST04  )                     (G)                                        (00890000)     5

                                                  ****************************
                                                  *  REQUEST NAME - MAIN     *
                                                  *  INPUT STREAM REQUEST    *
                                                  ****************************
-----------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                                  MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID            ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM       73 TO 80    SEQ
-----------------------------------------------------------------------------------------------------------------------------------
(ER)                                                                                                                  (00910000)     6
-----------------------------------------------------------------------------------------------------------------------------------
STMT SEQ  LOG CON .....OPERAND-A..... OPER  ............OPERAND-B.............  ......RESULT...... PARTIAL FIELD  SEQUENCE   LIST
TYPE NO. LEV CTR QLF FIELD   SEG-LVL  ATN   QLF FIELD   SEG-LVL  (OR CONSTANT)   QLF FIELD  SEG-LVL  STRT NUMB OPR  73 TO 80    SEQ
-----------------------------------------------------------------------------------------------------------------------------------
(PR)                                  (GO)     (SUB SERIAL                    )                                      (00910000)     7
(PR)                                  (GO)     (SUB DIRECT                    )                                      (00910000)     8
.
.
.

                                  REPORT: RHIGROUP                                                                    PAGE 1
-----------------------------------------------------------------------------------------------------------------------------------
  CUSTNO  CUST/ORD NO. RSTATUS
-----------------------------------------------------------------------------------------------------------------------------------
  00002    0000110001    KEQL
  00002    0000110004    KEQL
  00002    0000110005    KEQL
  00002    0000220003    KEQL
  00002    0000220005    KEQL
  00002    0000330001    KEQL
```

Vertical Ellipsis

# Understanding VISION:Builder

VISION:Builder is a general purpose software system for the design, implementation, and operation of data processing applications. It can be considered to be a higher level, non-procedural, results-oriented programming language above COBOL, APL, PL/I, and other procedural languages. As a result, simple statements in VISION:Builder perform complex file handling, database management, and report functions easily.

All the mundane tasks, such as file opening and file closing, I/O, data conversion, and error checking, are handled automatically by VISION:Builder. (There are over 2,000 automatic functions performed by VISION:Builder.) This leaves the application programmer free to design and test programs with little or no procedural concerns.

The file definitions are stored and retrieved from the VISION:Builder common library when needed at processing time. Changes to files do not involve changes to processing logic, making the application independent of the structure of the file.

The power of VISION:Builder can be seen through its file maintenance capability using transaction processing.❹ With only a few non-procedural statements, you can construct an application that reads a transaction file, loads a database, and outputs a report.

For more demanding requirements, transaction processing allows access to transactions before, during, and after updating files. This allows for extensive validation that can be automatic or user specified using simple non-procedural specifications or extensive procedural logic. Because transaction definitions (actions to be taken against the master file) are stored in the VISION:Builder common library, they become available across several applications that update the same file.

Reports are generated automatically by referencing fields. The fields are formatted automatically in columns across the report page. Numerous options are available, such as multiple image reports, editing output fields using patterns, dynamically modifying a report line before output, collating reports into alternate sequences, routing reports or subsets of a report to various destinations, and formatting your own report independent of VISION:Builder default formats, to name just a few.

VISION:Builder runs under the following IBM® operating environments:

| Operating Environment | Operating System |
|---|---|
| OS | z/OS<br>OS/390 |
| VSE | VSE/ESA™ |
| VM | VM/ESA® |

VISION:Builder is operating environment dependent (although VISION:Builder functions identically under the supplied operating environments); therefore, a change from one operating environment to another requires a new VISION:Builder environment. For more information on the interaction between VISION:Builder and the operating environment, see the *VISION:Builder for OS/390 Environment Guide* (OS/390 users) or the *VISION:Builder for VSE Environment Guide* (VSE users).

The VISION:Builder software system has several models grouped in two different series.

- The 4000 model series represents the VISION:Builder software system that allows users to develop fully functional applications that include the transaction processing and master file updating facilities along with the entire set of information retrieval, selection, and manipulation functions used in conjunction with the multitude of reporting and data extraction capabilities.
- The 2000 model series represents the VISION:Builder software system subset known as VISION:Two. This subset of VISION:Builder includes all the same information retrieval, selection, and manipulation functions that are used in conjunction with the multitude of reporting and data extraction capabilities that are part of the 4000 model series. However, the transaction processing and master file updating facilities are not included in the 2000 model series.

The term VISION:Builder is used to represent both the 4000 model series and the VISION:Two 2000 model series. The model number appears on the banner page that is output at the start of all source listing displays.

The VISION:Builder software system has three processing options that provide access to databases. The database options that are available are IMS, DB2, and GDBI. These options are available for both the 4000 and 2000 model series.

- The IMS option provides support for processing information in IBM IMS Databases using the standard DL/I processing facilities.
- The DB2 option provides support for processing information in IBM Database2 Relational Tables using the standard SQL processing facilities.
- The GDBI option provides a facility for interfacing user code with the standard mechanisms of VISION:Builder to perform I/O operations using any database processing facilities.

The options selected appear on the banner page that is output at the start of all source listing displays.

# Contacting Computer Associates

For technical assistance with this product, contact Computer Associates Technical Support on the Internet at esupport.ca.com. Technical support is available 24 hours a day, 7 days a week.

Chapter

# 2 Application Development System

VISION:Builder is a software system that can be used to design, implement, and execute data processing applications. Novice users can be productive within a few hours, yet VISION:Builder produces complex applications efficiently. An experienced user can build an application in VISION:Builder in less than half the time it would take for a similar application in COBOL or other third-generation languages.

The philosophy in using VISION:Builder is to:

- conceptualize how VISION:Builder functions during input, processing, and output
- design applications that use as many of the automatic functions of VISION:Builder as possible

VISION:Builder provides extensive facilities for the creation and maintenance of files and databases, as well as report generation. VISION:Builder processes virtually any data source and database format available on an IBM host (for example, flat files, IMS, and DB2) without requiring the programmer to be familiar with these formats.

The VISION:Builder system is made up of the following main components:


- VISION:Builder Engine on page 2-1
- VISION:Workbench for DOS on page 2-2
- VISION:Workbench for ISPF on page 2-3
- VISION:Builder Library Utilities on page 2-4.


## VISION:Builder Engine

The VISION:Builder engine is a versatile system for executing batch applications. It is a high level, non-procedural, fourth-generation system designed to handle applications from the simplest to the most complex. VISION:Builder performs complex file handling, database management, and report functions easily. All the

mundane tasks such as file opening and file closing, I/O, data conversion, and error checking are handled automatically. This allows you to design and test programs with little or no procedural concerns.

You do not have to be concerned with the source or structure of the data to be accessed. All accesses to a database are completely transparent. This means that once a data source is defined to the system (for example, IMS, DB2, and VSAM), reference to that data source in a program results in the automatic generation of the appropriate calls to the DBMS or access method. Once a data structure is defined to VISION:Builder, any reference to that structure uses the data definition for processing.

Additionally, VISION:Builder can access multiple databases with different access methods and structures within the same application. These access methods are completely transparent. For example, you can create DB2 tables from IMS databases without any procedural statements.

VISION:Builder also uses transaction processing❹ to further enhance its file maintenance capability. Once the transaction is defined, VISION:Builder can automatically create, change, and delete master file records without a single procedural statement.

Transaction processing allows access to transactions before, during, and after updating for extensive validation, which can be automatic or user-specified. Because transaction definitions (actions to be taken against the master file) are stored in the common library, they become available across all applications that might update the same file.

VISION:Builder increases productivity and helps less experienced users easily solve relatively complex problems by performing more than 2,000 functions automatically. You can concentrate on the heart of your application while VISION:Builder handles the routine work.

Automatic functions include database handling, data validation and conversion, database update, table searches, and report creation. Generation of automatic functions occurs at various stages of processing and can always be overridden.

VISION:Builder takes advantage of the fact that all applications have certain processing steps in common. It automates most of these steps through the VISION:Builder processing cycle, where most-likely actions are taken as a default. When using these defaults, applications that use little procedural code and depend on the default and automatic functions of VISION:Builder can meet a wide variety of objectives.

## VISION:Workbench for DOS

VISION:Workbench for DOS is an intelligent, user-friendly work station, which runs on personal computers (PCs). It provides host application development features combined with the ease of use, flexibility, and responsiveness of the PC. It

lets you write applications and reports quickly, with a minimum of training. It gives you instantaneous response to entries along with immediate validation. There is no waiting for a response from a remote computer.

Another feature of VISION:Workbench for DOS is the graphic representation of data structures. Instead of requiring you to enter text, you construct diagrams that graphically represent the data structure. These diagrams consist of one or more connected boxes, each box representing a segment or relational table in the structure.

VISION:Workbench for DOS also allows you to simulate your report output from a batch application without any host communication. Sample reports, including sample values for all fields and summaries, can be generated and displayed. You can look at a sample of the report and decide whether you are satisfied with how it looks before you send the application to the host for processing.

## VISION:Workbench for ISPF

VISION:Workbench for ISPF helps you create and maintain your VISION:Builder batch applications and definitions under TSO. Modeled after ISPF/PDF, it uses the same paneling and command entry structure. VISION:Workbench for ISPF gives you interactive syntax checking as you maintain and build your VISION:Builder applications.

With VISION:Workbench for ISPF, you can manage libraries that can be shared by many Computer Associates products. You can allocate and initialize new libraries, condense, dump, and restore existing libraries, and work with individual library members — all from your ISPF terminal, interactively and online. VISION:Workbench for ISPF lets you submit your applications directly to VISION:Builder for execution in foreground or batch.

VISION:Workbench for ISPF takes full advantage of all ISPF facilities. This includes the ISPF command structure, a complete step-by-step approach featuring menus and predetermined paths to application building or maintenance, as well as an extensive help facility and in-depth tutorial. The interactive validation performed by VISION:Workbench for ISPF checks the entered code each time you specify a command or press Enter. If incorrect entries are found, helpful diagnostic messages appear.

Using VISION:Workbench for ISPF, you can retrieve entire applications or selected application entities (such as file definitions, table definitions, procedures, and transactions) from one or more common libraries or ISPF/PDF libraries as if these libraries were concatenated. You can store your updated application in the same library, either replacing the old member or creating a new application entity.

# VISION:Builder Library Utilities

The VISION:Builder Library Utilities *component* (COMLIB) operates on the host in conjunction with the VISION:Builder Engine. It is used by the engine for interactions with a common library data set that contains cataloged definitions. A common library (M4LIBn) contains a variety of item types (such as file, table, array, transaction, procedure, and report) that have been validated and cataloged by VISION:Builder.

During VISION:Builder engine definition/maintenance runs, the COMLIB Library Utilities component is used to compile all the cataloged definitions, and update the common libraries. The common library *utility* (MARKUTIL) is used to initialize, backup, restore, copy, merge, and list the contents of your common libraries.

You only need to execute MARKUTIL once and then you can initialize, dump, restore, and condense the common library. Also within the same job step, you can transfer cataloged items from one common library to another.

The COMLIB Library Utilities component is called upon to retrieve any definitions required by VISION:Builder during the processing of your application. The COMLIB Library Utilities component is an integral part of VISION:Builder.

# Discipline of VISION:Builder

VISION:Builder adheres to a basic processing cycle derived from the specific inputs and outputs that are defined for an application. An application can be customized by introducing additional components into your application.

- Procedure — for more information about procedures, see Chapter 5, *The Application Cycle*.
- Report — for more information about reports, see Chapter 9, *Reporting*.
- Subfile (an output file that consists of a subset or extraction of data from one or more input sources within a VISION:Builder application) — for more information about subfiles, see Chapter 7, *Output File Processing*.

The processing cycle combines the automatic functions of the cycle (for example, reading files, coordinating files, and writing files) with the additional components of your application. Coordination is the process of synchronizing one or more secondary files to a primary file. These components interact with the basic processing cycle to provide traditional data processing functions for performing data selection and manipulation, report formatting, and secondary file output.

**Note:** The transaction cycle and sequential file coordination cycle are optional and need not be a part of each VISION:Builder application. These cycles are depicted in Figure 2-2 on page 2-6 and Figure 2-3 on page 2-7.

Multiple procedures are processed against the same input file record, ensuring optimum efficiency. VISION:Builder automatically ensures that all records are made available to all relevant procedures.

Figure 2-1 depicts the processing cycle of VISION:Builder. See Chapter 5, *The Application Cycle* for more information about the basic part of the cycle (without the transaction cycle and sequential file coordination cycle).

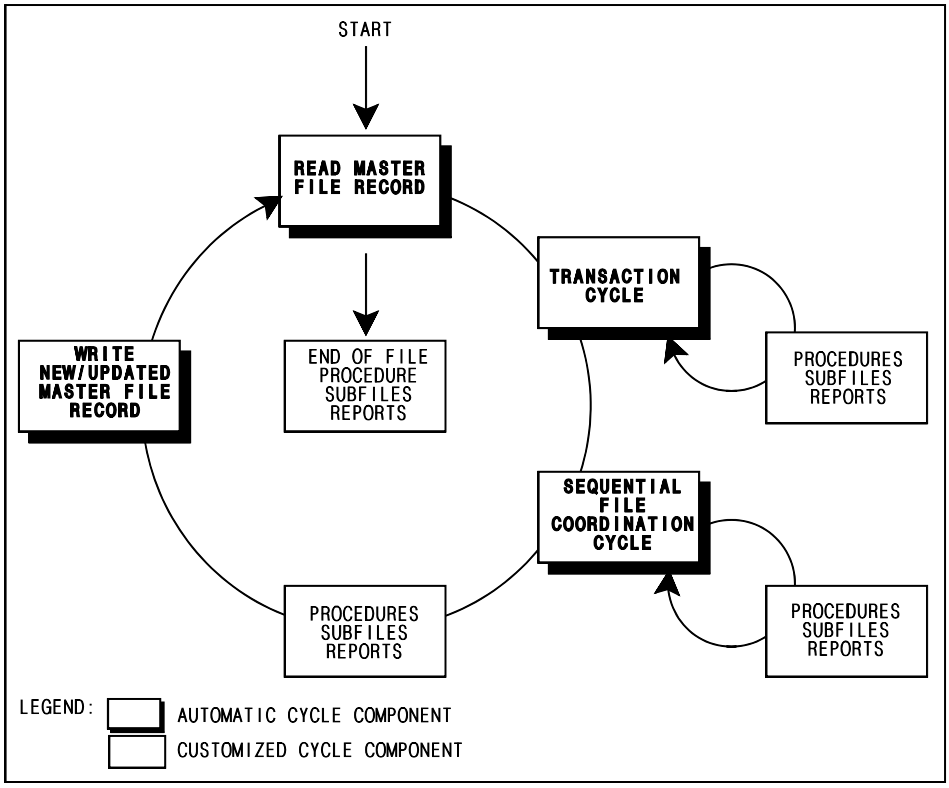



Figure 2-1 Processing Cycle

Figure 2-2 depicts the optional transaction cycle of VISION:Builder. The automatic components of the transaction cycle are described in Chapter 10, *Transactions*. The customizable components of the transaction cycle are described in Chapter 15, *Advanced Transaction Processing*.

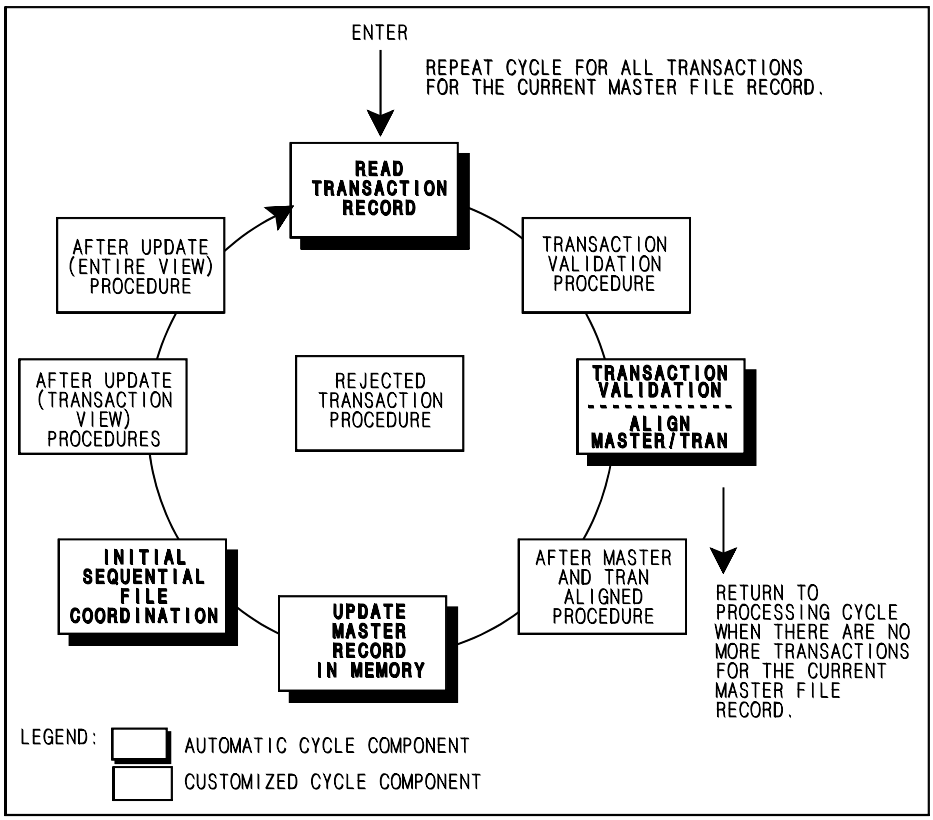



Figure 2-2 Transaction Cycle

Figure 2-3 depicts the optional sequential file coordination cycle. The coordination cycle is described in Chapter 6, *Input File Processing*.

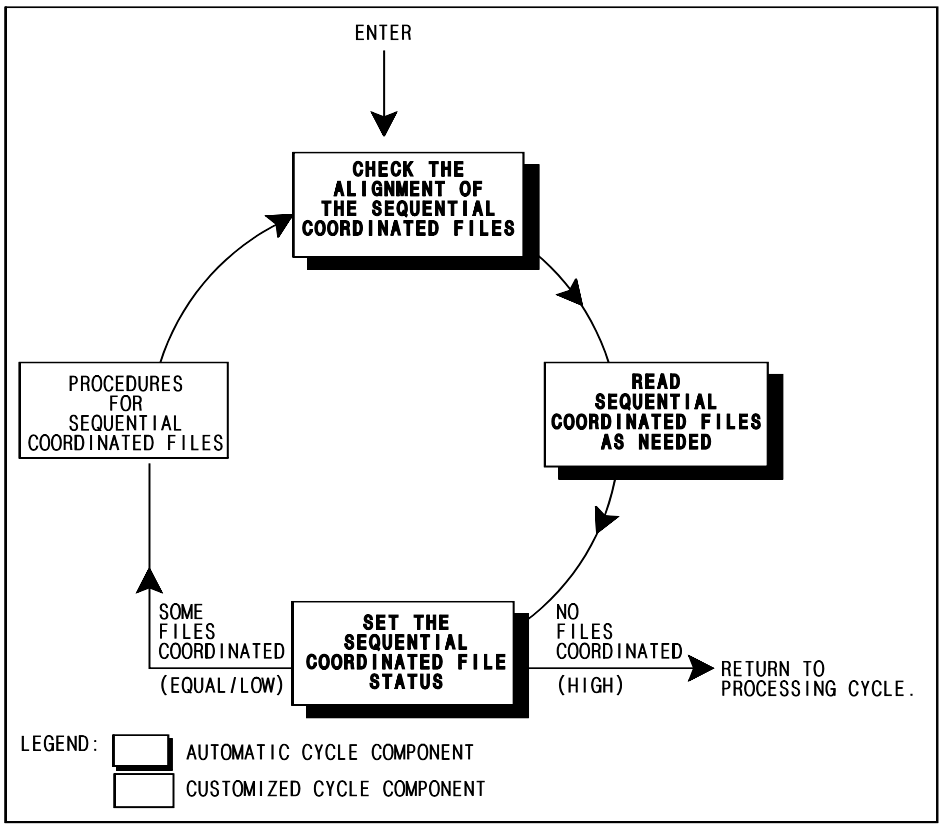



Figure 2-3 Sequential File Coordination

VISION:Builder begins its processing cycle by reading a record into memory. It makes all changes and extractions of data specified in the application. Completing the cycle, it outputs the changed data. For a complete understanding of the processing cycle, see Chapter 5, *The Application Cycle*.

The record in memory against which all the activity takes place is actually logical data (see Figure 2-4). VISION:Builder always operate against logical data. It is this capability that makes the VISION:Builder specifications and processing flow data-independent. Support of a database access method revolves around mapping the physical data into the required logical representation.

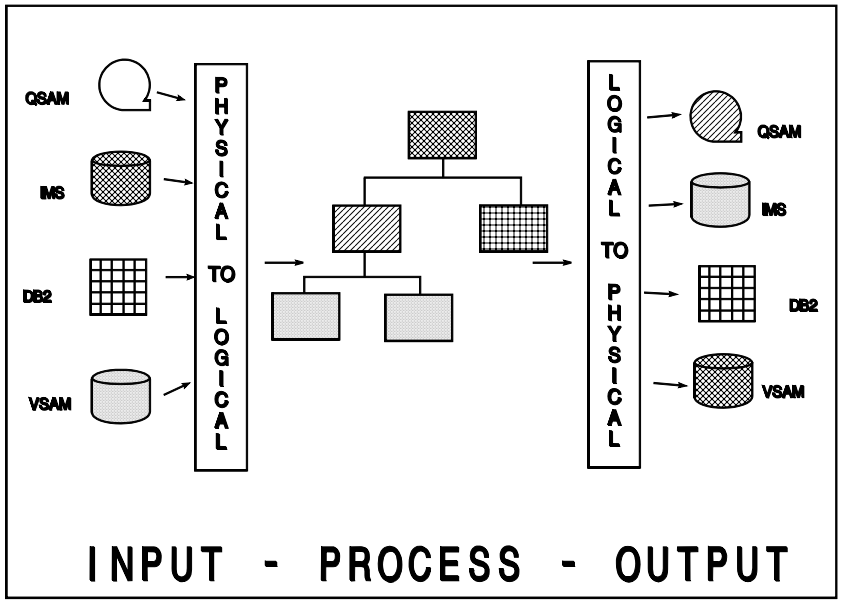



Figure 2-4 Logical Record vs. Physical Record

In some cases, the logical data may have the same structure as its physical counterpart on the disk, as when the file definition embraces a complete record managed by VSAM or by IMS without use of a secondary index. In other cases, portions of the physical data may not be present in the logical data, as when the file definition only describes a subset of segments in an IMS record or a subset of a DB2 table. The physical data can be rearranged, as when IMS secondary indexes are used. The logical data can be drawn from several different files, as when several joined DB2 tables are treated as one record.

## Automatic Functions

VISION:Builder performs over 2,000 functions automatically. These functions are performed by VISION:Builder at various stages of the processing cycle, namely input, application processing, and output.

The following is only a partial list of the automatic functions performed by VISION:Builder.

### During Input

- Reads and key validates all files.
- Validates all fields referenced.
- Validates transaction definitions.❹
- Allocates buffers and I/O areas.
- Reads relational, IMS, VSAM, and other database files randomly or sequentially.
- Updates segment count fields (if used).
- Lists transaction record errors.❹

### During Application Processing

- Checks for arithmetic overflow and invalid operations (for example, dividing by zero).
- Converts fields to other types specified.
- Aligns decimal places.
- Allocates buffers and work areas.
- Matches multiple input files for processing.
- Provides interrupt-handling routines.
- Provides over 1,500 diagnostic messages.

### During Output

- Produces new master files,❹ subfiles, and audit files.❹
- Formats and edits fields for reports.
- Produces glossaries, source statement listings, and reports.
- Prints file and storage statistics.
- Prevents invalid data from being generated.
- Produces reports in multiple formats.

These automatic functions will become clearer in the chapters that follow. An understanding of the automatic functions and the processing cycle of VISION:Builder is vital for building effective applications.

# Using VISION:Builder

There are two types of VISION:Builder runs: the definition/maintenance run and the application run. A definition/maintenance run is executed once to define to VISION:Builder the physical and logical structure of a given file and the transactions❹ that create or update it. Once these descriptions are cataloged into the common library, they are available to be used by any number of requests in application runs against that file definition.

Application runs create, update, and use the defined files according to rules provided by you in the file and transaction definitions❹ and VISION:Builder procedures.

## The Definition/ Maintenance Run

Files are defined to VISION:Builder by file definition statements. These definition statements are cataloged in the common library (M4LIB) during a definition/maintenance run so that they can be retrieved during application runs.

The common library (also known as the VISION:Builder dictionary) is a data set used for storing VISION:Builder definitions (file definitions, transaction definitions,❹ table definitions, array definitions, and procedures).

Every common library must be initialized through an initialization process that formats the library suitably. Initialization is done just once unless the entire library is to be deleted. The visible output of a definition/maintenance run is a VISION:Builder glossary that lists the fields and their attributes.

**Note:** Refer to the *VISION:Builder for OS/390 Environment Guide* (OS/390 users) or the *VISION:Builder for VSE Environment Guide* (VSE users) for more information about the JCL for the definition/maintenance run.

Figure 2-5 illustrates the definition/maintenance run.

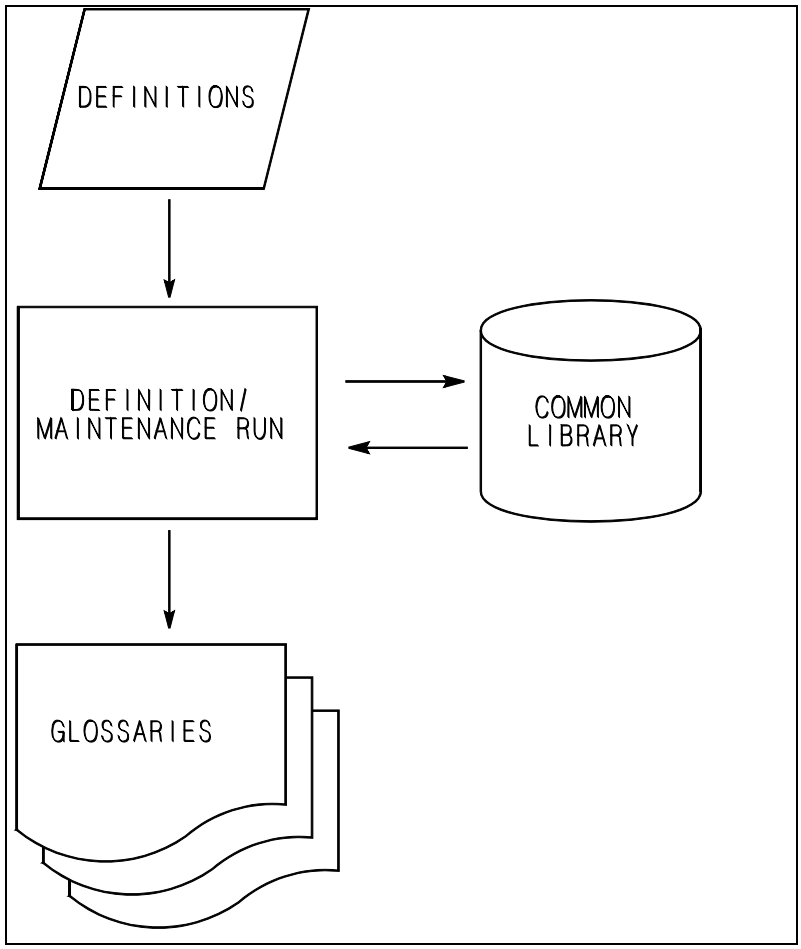



Figure 2-5 Definition/Maintenance Run

The definition/maintenance run can also be used to change or delete definitions from the common library. Additionally, information regarding definitions and procedures can be retrieved. Using the item tracking facilities, the following information can be retrieved:

- date/time cataloged or updated
- user data
- expiration date.

For further information regarding item tracking information retrieval, see Chapter 19, *Cataloging*.

The file and transaction❹ definitions are processed by VISION:Builder and the results are stored in the common library for later use in compiling application requests. Until such definitions exist, no procedures can be written against an existing file nor can a new file be created. The file definition itself does not create a

file; if it does not pre-exist, the file must be created during the application run. This is a logical file definition and each physical file could, in fact, have many logical definitions. A logical definition does not need to describe every field in the physical file record.

## The Application Run

Application runs usually contain three logical steps. These logical steps can be run as three distinct job steps or combined into a single job step. The first logical step is the processing step, which includes the compiling of the VISION:Builder source statements, reading of transaction❹ and/or master files, output of new files (including a new master file❹), and producing VISION:Builder report files.

The second and third logical steps are the sort and report steps. If report files are requested for an application, one or both of these steps are required.

For detailed information about single-step and three-step processing, refer to the *VISION:Builder for OS/390 Environment Guide* (OS/390 users) or the *VISION:Builder for VSE Environment Guide* (VSE users).

The application run is shown in Figure 2-6 on page 2-13. In this figure, a VISION:Builder file glossary and output requirements are used to write VISION:Builder application source statements for procedures, reports, and subfiles. This input combined with the appropriate job control language is submitted for execution.

### Processing Step

You must compile source statements before you actually process the master file. The source statements are read and checked for errors. The compiled source statements and file definitions are used as input to the VISION:Builder processing step. Once this step is entered, the master and transaction files❹ are passed against the system.

At the conclusion of the processing step, the master file is at end of file and all data generated or extracted from that one pass has been processed. All that is left are the steps needed to produce the printed output. The basic philosophy is that the master file is passed only once and every record in it is processed in turn.

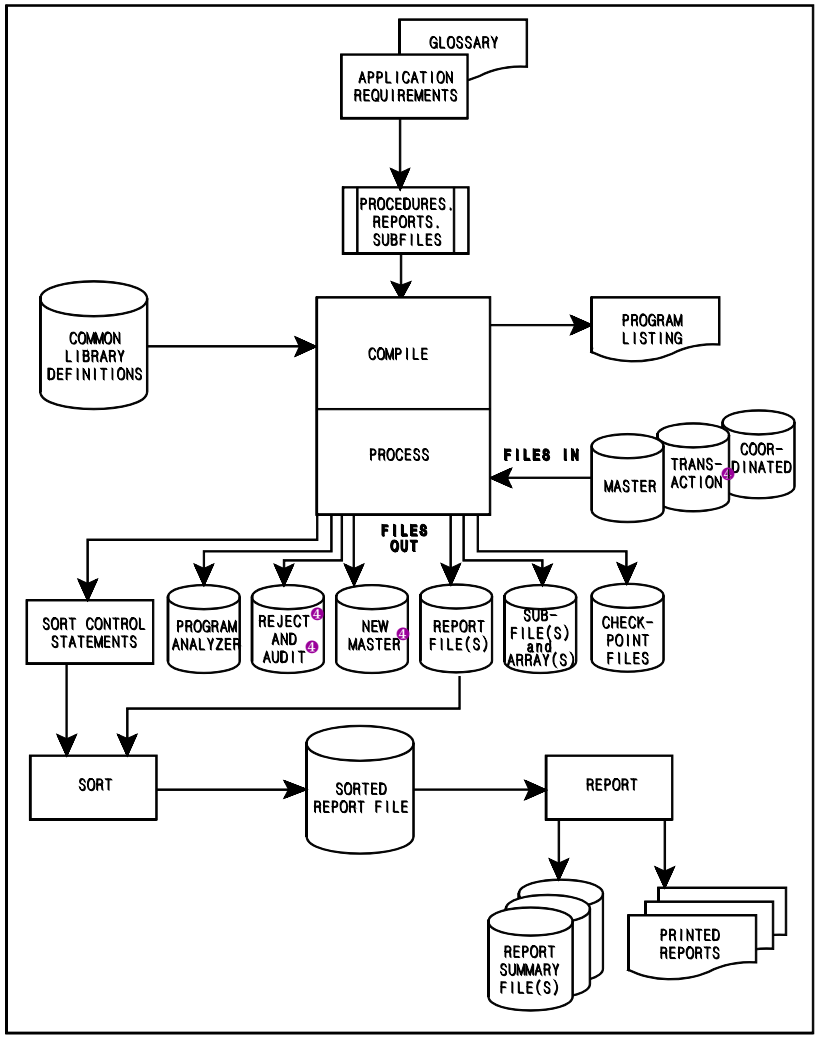



Figure 2-6 Application Run

## Sort and Report Steps

If multiple reports have been included in the application run, records selected by the criteria specified in those reports are processed and the output data for each report is placed on the report file. The result is a merged file whose primary sequence is the same as the input processing sequence. The secondary sequence is the order of reports. The report file also includes data fields used to sort the file to the order specified by the application request.

The number of procedures in any run against any file can vary from run to run. The lower limit is zero and the upper limit is primarily dependent upon the storage limitations imposed by machine or partition size. When there is more than one report or the sort sequence in the report is different from the input sequence, the

report file must be sorted before being processed by the report step. When sorting is required, a utility sort is used; VISION:Builder automatically generates the control statements that the sort requires.

The VISION:Builder report step generates the print line image for output. This step compiles the output of the processing step, provides proper carriage control, generates page titles and column headings, and calculates all specified summaries.

If requested, the report step creates report summary files that contain the calculated summary values, collates alternate report sequences, and routes the reports to various destinations. Report routing and report collating are described in Chapter 17, *Report Manager*.

# Input and Output Files

Up to 11 input files, 22 output files, and 255 report summary files can be processed by VISION:Builder in a given run. A wide variety of file structures can also be processed automatically, such as relational (for example, DB2), IMS, VSAM, and VISION:Builder arrays. Where other file structures are used (for example, CA-IDMS/DB, ADABAS), VISION:Builder provides an interface using external calls, GDBI, or user I/O. External calls are described in Chapter 18, *External Program Linkage* of this manual. GDBI is described in Chapter 3, *Defining Data* of this manual. Refer to the *VISION:Builder for OS/390 Environment Guide* (OS/390 users) or the *VISION:Builder for VSE Environment Guide* (VSE users) for information about user I/O .

## Input Files

Figure 2-7 illustrates the variety of input files processed by VISION:Builder.

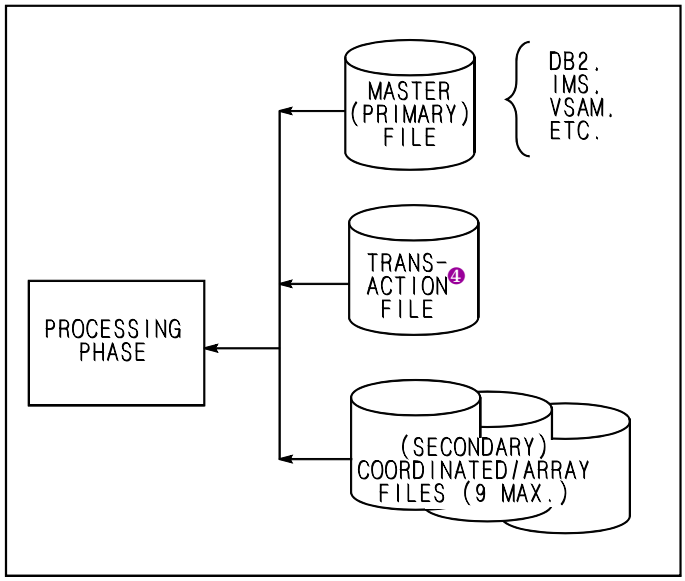



Figure 2-7 Input Files Processed by VISION:Builder

In total, there are up to 11 input files available to the processing step.

| | |
|---|---|
| (Old) Master File | File containing the basic source of data. It is the only file that can be directly updated. |
| Transaction File❹ | Group of transaction records used to add, delete, or modify data on a master file according to the transaction definitions for that master file. |
| Coordinated Files | Files used as auxiliary input and usually related to the master file. There are several means by which coordinated files can be processed: sequentially, randomly, user-read, or in parallel (described in detail in Chapter 6, *Input File Processing*). |
| Array Files | Files logically processed using two dimensions, rows and columns (described in detail in Chapter 13, *Arrays*). It is an array file format unique to VISION:Builder. |

## Output Files

Figure 2-8 illustrates the range of output files that a VISION:Builder run can produce.

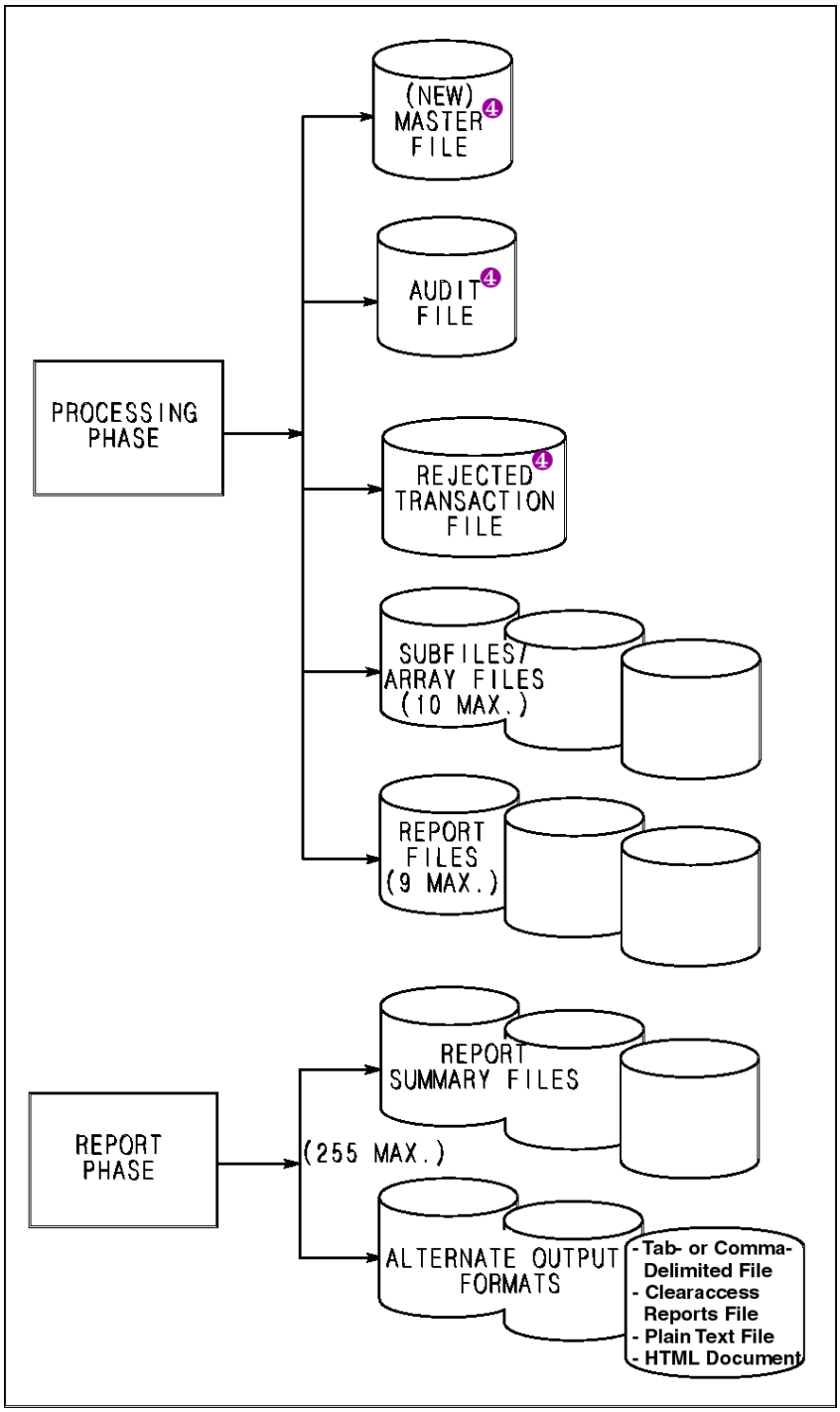



Figure 2-8 Output Files Processed by VISION:Builder

In total, a given run can produce up to 22 output files: new master, audit, reject, 9 report files, and up to 10 subfiles and/or arrays. A maximum of 255 report summary files can be produced, for any or all reports in the run.

| | |
|---|---|
| (New) Master File❹ | Copy of the (old) master file that reflects the additions, deletions, and modifications of data due to transactions and application procedures. This file is utilized when the (old) master file is not directly updated-in-place. |
| Audit File❹ | Optional output file containing master file records deleted during a run (see Chapter 5, *The Application Cycle*). |
| Rejected Transaction File❹ | Optional output file containing transaction records rejected during a run (see Chapter 10, *Transactions* and Chapter 15, *Advanced Transaction Processing*). |
| Subfile | Output file containing data selected or generated in an application run and not necessarily intended to be used to generate a report (see Chapter 7, *Output File Processing*). |
| Report File | File in internal VISION:Builder format that contains data selected for output to a report (see Chapter 9, *Reporting*). |
| Array File | Output file containing data intended for use as an input array file in a subsequent application run (see Chapter 13, *Arrays*). |
| Report Summary File | Optional output file containing the calculated summaries and grouping information for any or all reports in the run (see Chapter 7, *Output File Processing*). |
| Tab-Delimited File | Contains data fields selected for output as a tab-delimited file (see Chapter 7, *Output File Processing*). |
| Comma-Delimited File | Contains data fields selected for output as a comma-delimited file (see Chapter 7, *Output File Processing*). |
| VISION:Clearaccess™ Report File | Contains data fields selected for output as a VISION:Clearaccess reports file (see Chapter 7, *Output File Processing*). |
| Plain Text File | Contains a report selected for output as a plain text file (see Chapter 7, *Output File Processing*). |
| HTML Document | Contains a report selected for output as an HTML document (see Chapter 7, *Output File Processing*). |

# Transaction Processing❹

Transaction records or transactions in VISION:Builder are action-oriented records that maintain or update files. Transaction definitions are defined and cataloged in the common library. They are automatically retrieved and applied to the input master file during application processing (see Chapter 10, *Transactions* and Chapter 15, *Advanced Transaction Processing*).

VISION:Builder handles the I/O to all input and output files automatically. Virtually any type of file organization is supported, including relational, IMS, VSAM, SAM, ADABAS, CA-IDMS/DB, and other database organizations.

The master file is the primary input file to VISION:Builder. It is automatically updated by transactions. When the master file is processed (or updated) sequentially, it is the driving file for a particular application run. When transactions update the master file directly (randomly), the transaction file is the driving file. The driving file initiates the sequence of events that comprise the application run.

Figure 2-9 is a copy of Figure 2-1 with the transaction cycle area highlighted. Figure 2-10 details the transaction cycle in VISION:Builder.

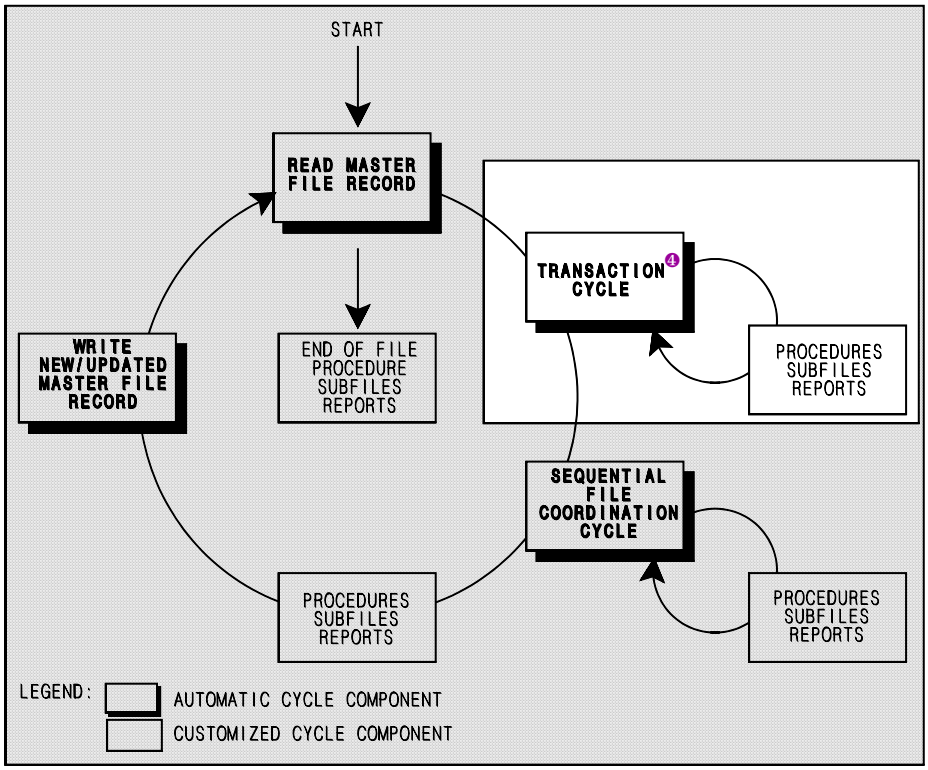



Figure 2-9 Processing Cycle

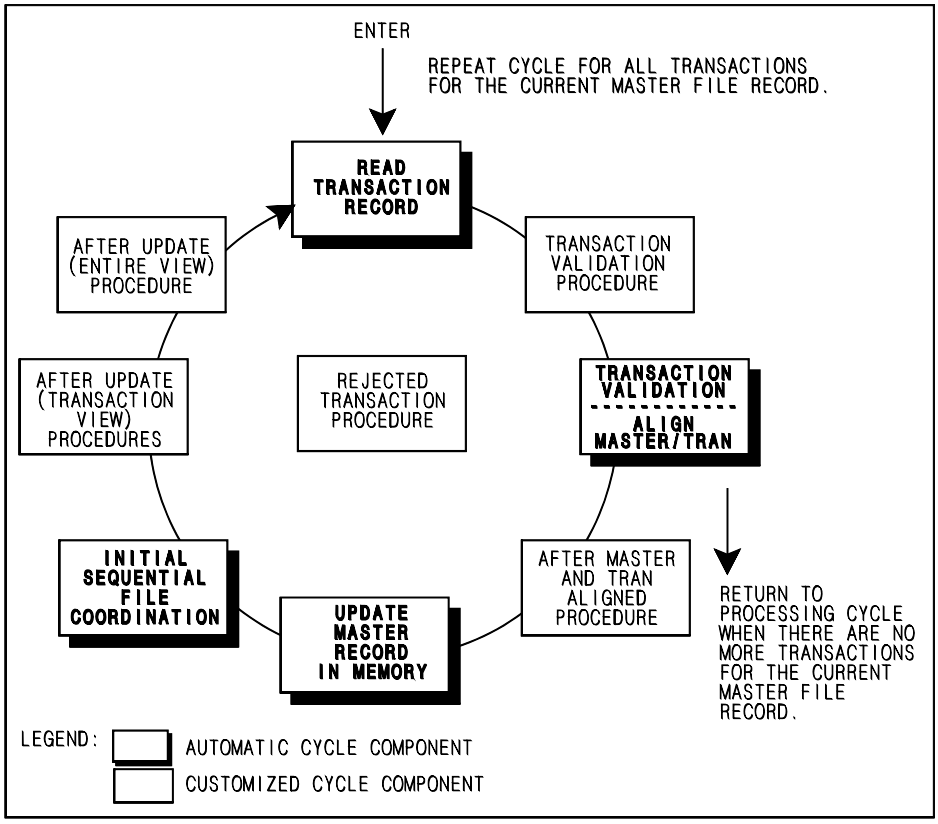



Figure 2-10 Transaction Cycle

VISION:Builder automatically reads the first transaction record from the transaction file and automatically validates it (see Chapter 10, *Transactions*). Assuming the transaction record was not rejected by VISION:Builder, the first master file record is automatically read. The master file record is aligned with the transaction record by comparing the key values of the records. If the transaction record key is less than or equal to the master file record key value, the transaction record automatically creates the master file, in memory, using the transaction definition from the common library. Any customizing procedures present are executed. The next transaction record is read and processed against the same master record.

In this manner, several transaction records can be used to update a single master record. When the transaction record key value exceeds the master record key value, the master record (or updated record) is available for processing by the application procedures in the processing cycle. See Figure 2-1 on page 2-5 for information about the processing cycle. See Chapter 5, *The Application Cycle* for a discussion of application procedures. The application procedures can choose to produce a report, a subfile, or neither. Once the master record is processed by the application procedures, the new (updated) master record is automatically written to a file, if specified.

The important concepts to note are that the input files are automatically read and processed in a specific way by VISION:Builder. You, as the application programmer, need only concern yourself with the application procedures, bearing in mind the manner in which the input file records are processed.

The application processing flow will be expanded and described in greater detail throughout this manual.

## Tables

VISION:Builder tables are easy to use and provide extensive capabilities beyond seeking simple match conditions, such as retrieving results based on closest, higher, lower, or range values around a table argument. Tables can be of binary, sequential, and displacement type (categorized by the search process used).

A VISION:Builder table consists of two parts: a column of table argument values and a column of respective result values. All tables must be defined and cataloged in the common library using a definition/maintenance run. Once defined and cataloged, tables can be used procedurally or automatically in any procedure, report, or subfile specification.

Procedural table lookup requires looking up the table using a procedure statement and an input argument. This input argument can be a field in any work area except variable length fields. The field values are used as input to the table lookup and the selection criteria (equal, nearest, and greatest) between the input argument and table argument is applied. If found, the result value is retrieved. The result field can be selected at application run time using the LOOKUP function. For more information about the LOOKUP function, see the *VISION:Builder ASL Reference Guide*.

Automatic table lookup requires looking up the table by referencing a predefined result field. The result field, argument, and table are specified in the file definition and cataloged in the common library. At processing time, the result value is automatically returned to the application, transparent to the programmer. For more information about tables, see Chapter 12, *Tables*.

## Arrays

Arrays are useful for processing large amounts of data in memory, especially for performing calculations on rows and columns of data and reporting their results without the detail.

An array record consists of data, logically defined along two dimensions, rows and columns. The intersection of a row and a column is called a data cell. Data cells contain one occurrence of one or more fields. All data cells comprise the same set of fields.

For example, ARRAYAB consists of two rows and three columns and two fields FLDA and FLDB. FLDA has data values A1, A2, A3, A4, A5, and A6; FLDB has data values B1, B2, B3, B4, B5, and B6 (see Figure 2-11*)*. If a key exists, it is at the beginning of the record and is not part of the data cells. For more information about arrays, see Chapter 13, *Arrays*.

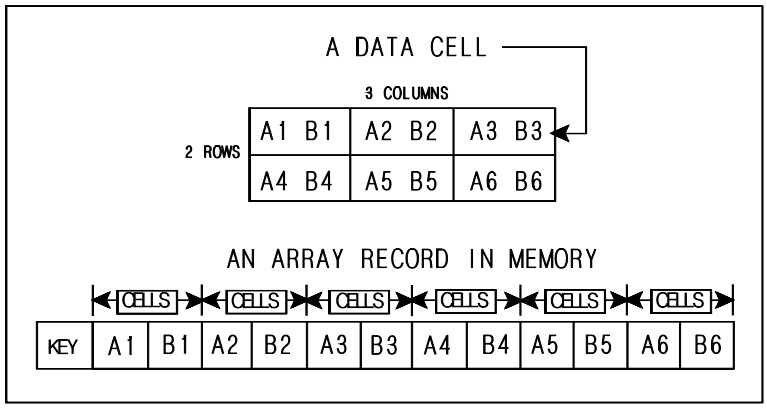



Figure 2-11 Array Record Structure

# Reporting

VISION:Builder produces a standard report with many format defaults, some of which can be easily adjusted or overridden. Others are completely standardized as to location and content.

The layout of the report page follows a standardized format. The title lines are followed by subtitles, column headings, report detail lines, and summary information. Standard reports also comply with certain rules. For example, title lines can contain only literals and not data fields.

If you want to customize your report in a way not provided for by the overrides in a standard report, VISION:Builder offers a number of different options. These options are useful when reports require data values in the title or column headings or when custom labeling and placement of summaries is wanted. In addition, you can also perform arithmetic calculations on the summaries that VISION:Builder has computed and report the results in the summary section.

If the standard report format is acceptable but you want to customize just the report title, column headings, or summary areas, VISION:Workbench for DOS provides you with the capability of painting the report sections you want while accepting the standard report layout for the rest of the report.

In the fixed format syntax of VISION:Builder, you can specify a formatted sectional report to customize just a section of a standard report or you can specify a formatted report in which you control the report layout and include your customizations.

Dynamic Report Line Modification offers another method to enhance the format of a report.

All VISION:Builder reports (both standard and customized) can contain different types of summaries (for example, total, count, average) on any fields you specify. VISION:Builder automatically computes these summaries for each group you specify. You can specify up to nine levels of grouping fields, plus grand summaries.

Collating and routing reports are features of the Report Manager function. Using the collating function, you can control the order in which reports are printed, collate subsets from a collection of reports, and print the subsets in a specified sequence.

Using the routing function, you can tap the full facilities of your operating system and its communications network by routing reports and report subsets to remote sites, microfiche devices, special forms, multiple copies, tape, disk, and so on.

For more information about reports, see Chapter 9, *Reporting*, Chapter 14, *Extended Reporting Capabilities*, and Chapter 17, *Report Manager* of this manual.

Chapter

# 3 Defining Data

The fundamental principle behind VISION:Builder data definitions is to allow applications data independence by separating the description of the data from the physical data. VISION:Builder uses the concept of structured data, which allows full use of many automatic functions, such as automatic looping through segments of data.

VISION:Builder easily deals with many types of data structures, such as relational tables, IMS hierarchical records, VSAM, and data structures from other database management systems. Data from any of these sources can be mapped to a VISION:Builder data structure. Thereafter, VISION:Builder can present you with logical data views of the data. For example, VISION:Builder will automatically join multiple relational tables, presenting you with what looks like one logical record. For more information about relational files, see Defining a Relational File on page 3-42.

The data definitions are easy to create using VISION:Workbench for DOS. Once the data is defined, you export the definitions to the host, where you catalog and maintain them in the common library. Definitions can also be created directly on the host, using the VISION:Workbench for ISPF component of VISION:Builder.

Data definitions are usually made once and cataloged in the common library. Common libraries can be shared by many Computer Associates products. For more information about the common library, see Chapter 19, *Cataloging*. After the definitions are cataloged, applications from multiple products, both batch and interactive, can use the definitions.

Although this chapter discusses the concept of a hierarchical record structure at great length, it should be noted that VISION:Builder can be used equally well for applications using a simple (flat) file whose records do not contain any inherent structure. The examples in this chapter usually depict a definition containing a hierarchical structure of segments. A definition that contains only one segment is permissible and appropriate for many applications. These root-only definitions are prepared by defining only a single data segment and omitting all other structure specifications.

# Concept of Structured Files

VISION:Builder processes a wide variety of files with various formats and record structures. For example, acceptable file organizations include VSAM, SAM, and others. VISION:Builder also handles various database access methods such as DB2, IMS, CA-IDMS/DB, ADABAS, and others.

VISION:Builder can map relational tables into VISION:Builder structures. This chapter will use relational tables when explaining the concept of structured files. For example, you might have four relational tables: a Customer table, an Order table, a Shipped-Invoices table, and an Ordered-Items table. All of these tables are related through the CUSTNO or ORDERNO columns. You can define these to VISION:Builder as one structure and have VISION:Builder automatically and transparently join the tables.

The sample tables will map to the VISION:Builder structure as shown in Figure 3-1.

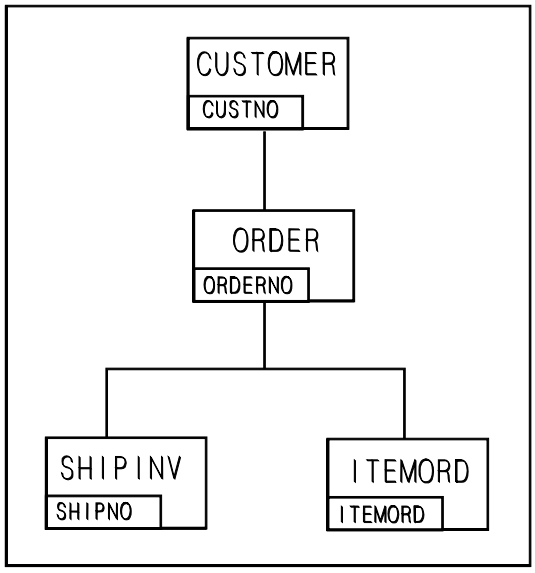



Figure 3-1 Logical Record Structure of Relational Tables

Each segment consists of a collection of fields. CUSTNO and all its related customer information (for example, CUSTNAME, and CUSTPH) make up a segment. Each segment has one field that uniquely identifies it. This field is called the segment key. CUSTNO is the segment key for the customer segment. A segment key field must be defined for every segment in the file.

In the Workbench, the hierarchical structure is defined graphically in the File Definition window. See Defining a Structured File to VISION:Builder on page 3-6 for more information. However, in the fixed format syntax of VISION:Builder, the position of a segment in the hierarchy is determined by assigning a segment number and a level number.

In Figure 3-2, the CUSTOMER segment is the root segment and is always given a level number of 1. Because there are many orders for each customer, the ORDER segment is given a level number of 2, one greater than the root. Similarly, the INSTALL segment is given a level number of 2. Each order can have many shipment numbers; therefore, the SHIPINV segment is given a level number of 3 (greater than number 2). Level numbers are assigned similarly for the ITEMORD segment (level 3) and the ITEMSHIP segment (level 4). Level numbers must be assigned consecutively.

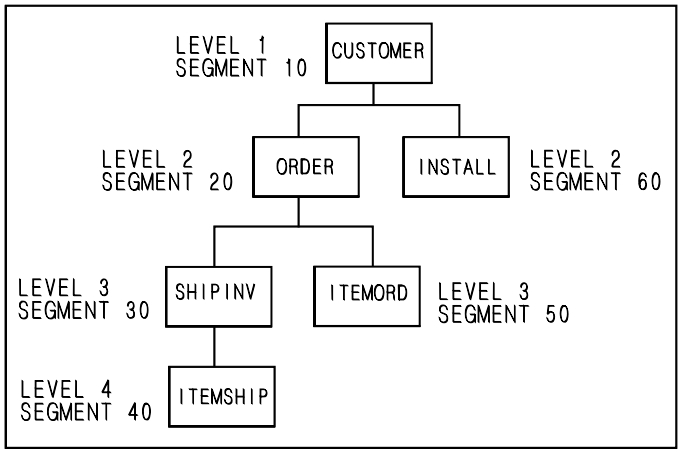



Figure 3-2 Level and Segment Numbers

Segments are numbered from top to bottom and left to right following a single path of the structure. The path identified by the segment types CUSTOMER, ORDER, SHIPINV, and ITEMSHIP is one such path. CUSTOMER, ORDER, and ITEMORD are also considered a single path.

In this hierarchy, the CUSTOMER segment is given the lowest segment number, in this case a 10. Traveling down the leftmost path, ORDER is given a segment number of 20 (greater than 10). SHIPINV is given a segment number of 30. ITEMSHIP is given a segment number of 40. Moving back to the immediate parent and down to its dependent segment, ITEMORD is given a segment number of 50. Moving back to the root and traveling down the rightmost leg, the INSTALL segment is given a number of 60 (greater than 50).

Files with fixed occurring segments, common in COBOL definitions, can be easily mapped to VISION:Builder. A file containing the root segment CUSTOMER, plus the ORDER segment occurring 44 times, is the equivalent of the COBOL definition of:

```
01CUSTOMER-RECORD
05 CUSTNO PIC (05).
05 CUSTNAME PIC (30).
05 CUSTPH PIC (10).
  .
  .
  .
05 ORDERSEG OCCURS 44 TIMES.
  10 ORDERNO PIC (05).
  10 ORDERDATE PIC (06).
```

```
10 ORPERSON PIC (03).
  .
  .
  . (ETC.)
```

Using structured files reduces the need for managing repetitive information in separate fields. It also reduces the need for managing several types of records, some containing fixed fields and others variable. The type of records you have to manage is generally limited by the database management system.

Figure 3-3 shows how VISION:Builder builds a structured record in memory from the hierarchy just defined. Customer A1 has orders B1 and B2. Order B1 has two shipping invoices, C1 and C2. Order B1 also has two items, E1 and E2. Shipping invoice C1 has three items shipped, D1, D2, and D3. Customer A1 also has two installations, F1 and F2.

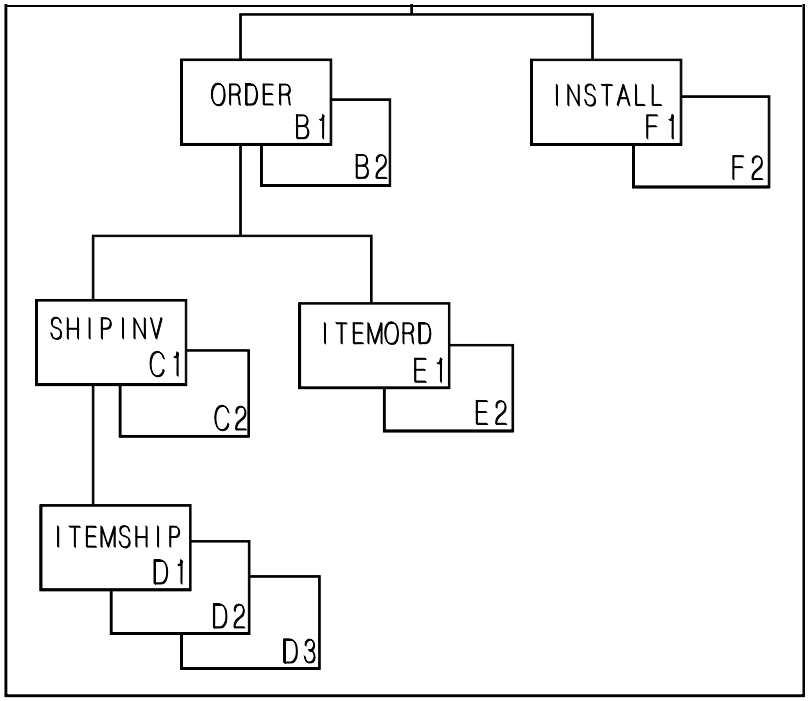



Figure 3-3 Segment Occurrences for Customer A1

Figure 3-4 shows how VISION:Builder builds a structured record in memory for each root key.

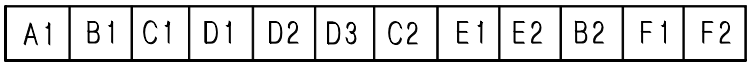



Figure 3-4 Structured Record in Memory

Each structured record is presented to each VISION:Builder procedure. Multiple procedures can act against one master file record. Procedures can reference one or more fields in lower level segments. Each executable reference to such fields starts a loop, allowing you to access the information in every segment occurrence. This function is described in Chapter 8, *Data Structure Handling*.

## Important Points to Note About Hierarchical Structures

- All files are considered structured by VISION:Builder. In its simplest form, however, a structure can consist of only one segment.
- Every segment must have a key.
- Segment numbers must be numbered top down, left to right, in ascending order. You do not need to specify this when entering the file definition using VISION:Workbench for DOS or VISION:Workbench for ISPF.
- Levels are numbered consecutively from top to bottom in ascending order. You do not need to specify this when entering the file definition using VISION:Workbench for DOS.
- Segments in a file are always of fixed length unless they contain a variable length field.
- For fixed length segments, the last byte of the segment must be defined whether or not its data is used.
- For variable length segments, the last byte of the fixed portion of the segment must be defined whether or not its data is used. The variable length fields start at the location following the last byte of the fixed portion of the segment.
- For variable occurring segments, a special count field must be defined in the parent segment unless the format is IMS, relational, or GDBI (mapped).
- Fields can be multiply defined and overdefined.
- Unless otherwise specified, the files are read serially and VISION:Builder assumes that the records are in ascending root key sequence. However, the files can also be read randomly, if specified.

## Terms of Hierarchical Structure

| | |
|---|---|
| Dependent | A segment in the hierarchical structure that is at a lower level than (below) another segment. For example, a segment at level 2 is said to be a dependent of the segment at level 1. |
| Field | The smallest addressable unit of data is a field. |
| Fixed Length Record | A record in which all segment types are fixed occurring and no variable length fields are defined. |

| | |
|---|---|
| Fixed Occurring | Segment type in the hierarchy that contains the same number of segment occurrences for each parent key. For example, if every customer had exactly ten orders, the ORDER segment would be called fixed occurring. |
| Level Number | Levels are horizontal layers of dependency. Every segment is assigned a level number. The lower the number, the higher the level. The root has the lowest number, always 1. |
| Parent & Child | Segment types immediately above and below a given segment type are called parent and child, respectively. |
| Path or Leg | A path or leg of the structure is the set of dependent segment types that starts with the root and contains one segment type at each level. |
| Segment | A group of related fields. |
| Segment Key | One field that uniquely identifies a segment. |
| Segment Occurrences | Number of occurrences of a segment. For example, if a customer has three orders (identified by the segment key ORDERNO), there are three segment occurrences of the ORDER segment for that specific customer. Segment occurrences exist only in relation to a specific segment type. |
| Segment Type | All the fields belonging to a segment are said to belong to a specific segment type. The segment type is identified by a unique number called the segment number. |
| Variable Length Record | A record in which any segment type is variable occurring or contains variable length fields. |
| Variable Occurring | Segment type in the hierarchy that contains a different number of segment occurrences for each parent key. |

## Defining a Structured File to VISION:Builder

Structured files are easily defined to VISION:Builder using VISION:Workbench for DOS. The File Definition window of the Workbench helps you visualize the structure of your file and create documentation for your program. For detailed instructions on each entry and on all the choices available to you, refer to the VISION:Workbench for DOS HELP windows or the *VISION:Workbench for DOS Reference Guide*.

Figure 3-5 shows the File Definition window.

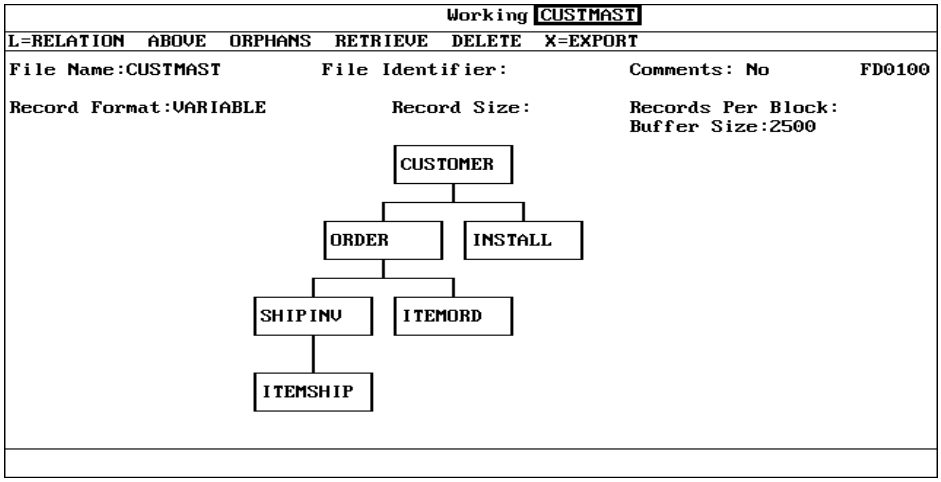



Figure 3-5 File Definition Window

The Segment Definition windows, one of which appears in Figure 3-6, define each segment's characteristics, fields within the segment, field characteristics, identify key fields and count fields for dependent variable occurring segments, column headings for the fields, and output edit information.

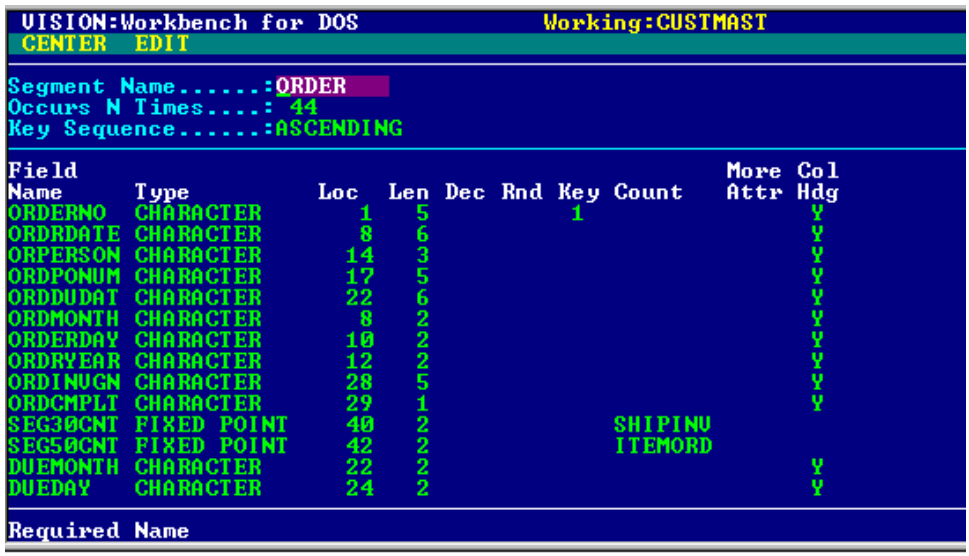



Figure 3-6 Segment Definition Window

Note that segments can be ordered in ascending or descending order using the Key Sequence entry.

Also note that fields within a segment can override other fields in the same segment. For example, the fields ORDMONTH, ORDERDAY, and ORDYEAR overdefine the field ORDRDATE in the ORDER segment shown in Figure 3-6. Overdefinition of fields is indicated by specifying the same or overlapping locations to two or more fields.

For variable occurring segments, a Count entry must be defined in its parent. For example, the field INSTCNT is defined in the root for the variable occurring dependent INSTALL segment. VISION:Builder uses the Count entry for automatically determining the number of occurrences of the segment.

For fixed occurring segments, the number of times the segment occurs must be specified in the Occurs N Times entry. For example, the ORDER segment occurs 44 times (see Figure 3-6).

The minimum buffer size is computed by taking the customer with the most number of orders, installations, items, and so on, and adding up the length of each segment (sum of its field lengths) and multiplying it by the number of occurrences. This should not exceed 32,760 bytes except for spanned VSAM, GDBI, relational structures, and IMS.

## File Glossary

The glossary is obtained through the Utility component of VISION:Workbench for DOS. A partial glossary is shown in Figure 3-7 and Figure 3-8.

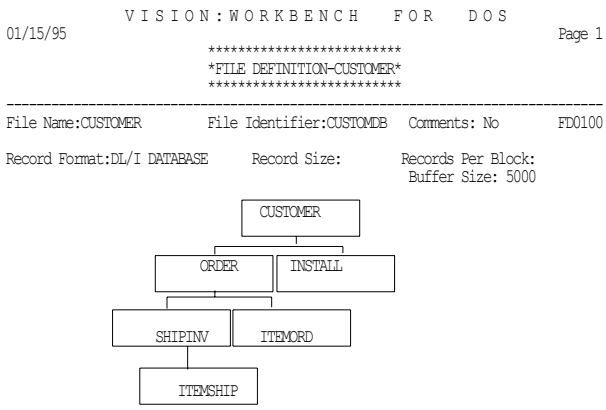



Figure 3-7 File Glossary – Definition

```
                         V I S I O N : W o r k b e n c h
07-31-97                                                                  Page 2

                              *************************
FILE NAME  -  CUSTOMER        *   SEGMENT - CUSTOMER   *
                              *************************

--------------------------------------------------------------------------------

Segment Name......:CUSTOMER           Suppress Dup....:                  FD0110
Occurs N Times....:                  Authorization ID:
Key Sequence......:                  DB2 Table Name..:


Field                                                          More Col
Name      Type          Loc  Len Dec Rnd Key Count   Attr Hdg
-----     ----          ---  --- --- --- --- -----   ---- ---
CUSTNO    CHARACTER       1   5           1
CUSTNAME  CHARACTER       6  30           S
CUSTPH    CHARACTER      36  10
```

Figure 3-8 File Glossary – Printout

A partial glossary prints out one page of information for each segment. Figure 3-8 only shows one segment. A full glossary also includes column headings, as well as other field-specific information. You can write or send the information to a printer or to a file.

# Entering a File Definition

This section explains how to define the Customer file described earlier using VISION:Workbench for DOS.

After logging on to your PC and invoking VISION:Workbench for DOS, the main menu appears, as shown in Figure 3-9.

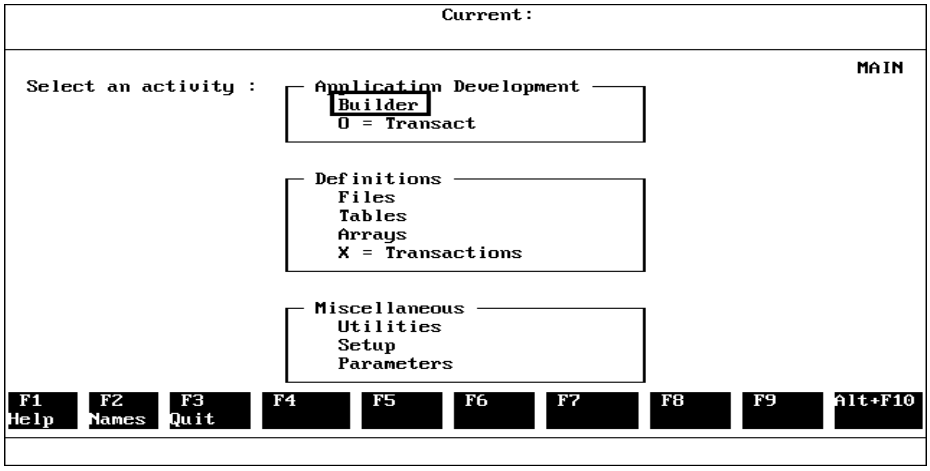



Figure 3-9 VISION:Workbench for DOS Main Menu

The Workbench has detailed HELP windows. Figure 3-10 shows the HELP windows for the Files option. To enter HELP, press F1. To exit HELP, press F3.

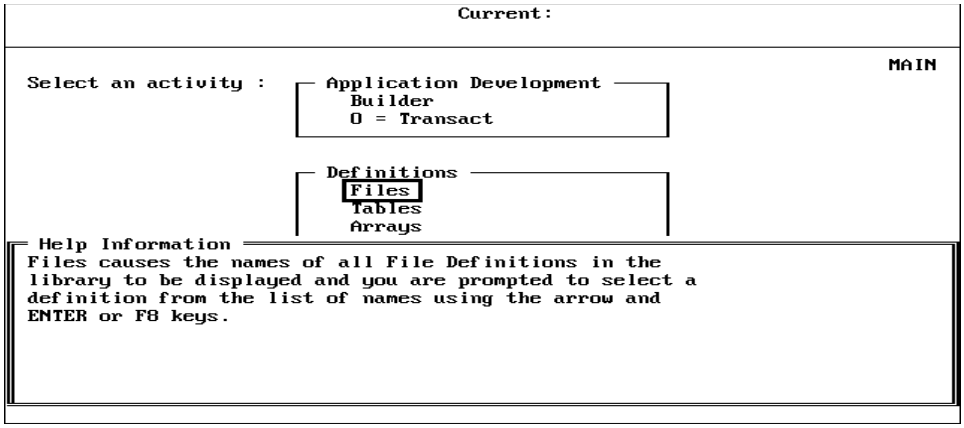



Figure 3-10 HELP Window for the Files Option

After exiting HELP, press F8 to proceed to the File List window to specify the name of your file (CUSTOMER). VISION:Builder automatically enters the definition type (FILE) (see Figure 3-11).

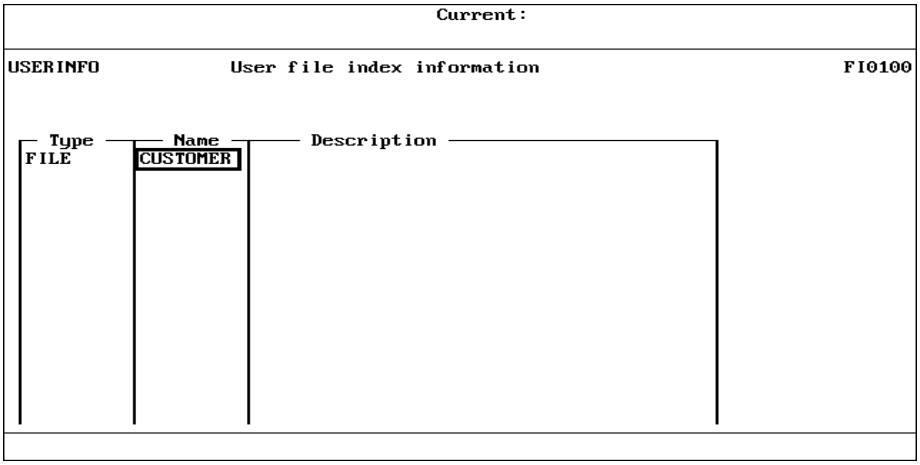



Figure 3-11 File List Window

Once the file has been named in the File List window, press F8 to open the File Definition window (see Figure 3-12).

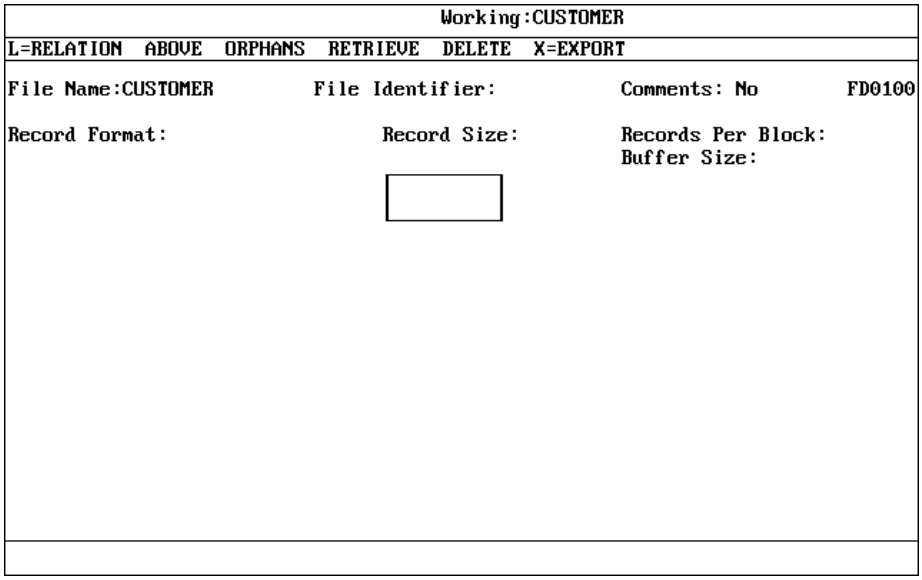



Figure 3-12 File Definition Window

The File Identifier entry for IMS files should contain the DBD name. For all other files, leave the entry blank.

The Comments field contains YES or NO depending on whether or not you have supplied any comments. To use the Comments feature, type YES in the Comments field or press F8 while the Comments field is highlighted. The Comments window will be displayed. Type in your comments. Press F3 to exit the Comments window.

Move the cursor to the Record Format entry. VISION:Builder supports numerous formats.

- IMS Database
- IMS HDAM
- Relational
- Key-Seq VSAM
- Entry-Seq VSAM
- VSAM Alt Index
- Fixed
- Variable
- Undefined
- ISAM
- ISAM Variable
- Compressed

The CUSTOMER file contains variable occurring segments, so give it a record format of variable.

The Record Size entry depends on the record format.

- For fixed length files and ISAM files, record size should have the number of bytes in the record.
- For unblocked variable length files, record size should contain the maximum expected record size, excluding the 4 bytes for the RDW (record descriptor word).
- For blocked variable files or variable ISAM files, record size is the maximum record size, excluding the 8 bytes for the BDW (block descriptor word) and RDW. If left blank, VISION:Builder computes it from the buffer size specification.
- Record size should be blank for file types:
  - IMS Database
  - IMS HDAM
  - Entry-Seq VSAM
  - Key-Seq VSAM
  - VSAM Alt Index
  - Relational
  - Undefined

The Buffer Size entry specifies the amount of storage required for blocked variable length or undefined records. The VISION:Workbench for DOS online HELP window explains the entry in detail.

To activate true VBS (variable blocked spanned) processing, code a maximum record size of 32K or larger in the Buffer Size entry and leave the Record Size entry blank.

VISION:Workbench for DOS has already given you a box for your root segment. Position your cursor within the box and type in CUSTOMER as shown in Figure 3-13.

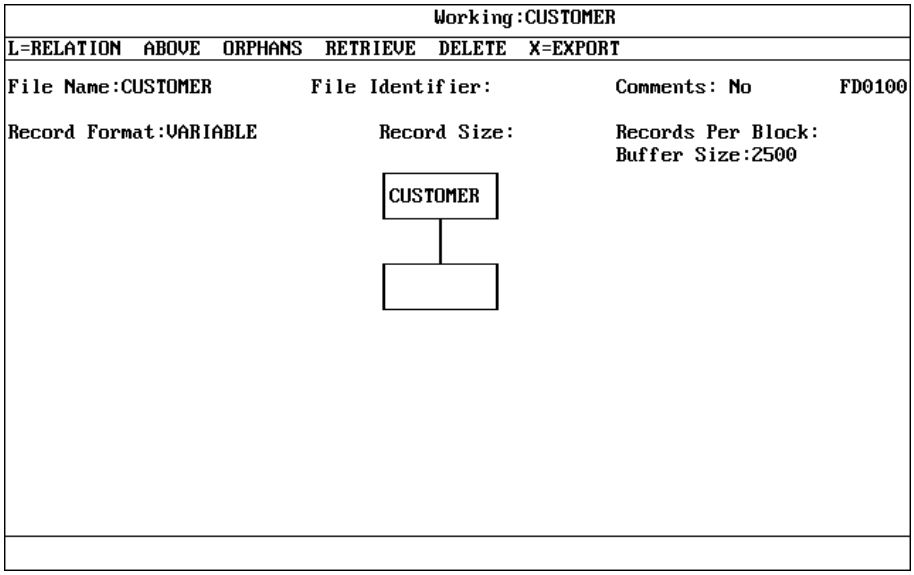



Figure 3-13 File Definition Window with One Segment

Because CUSTOMER fills in all available spaces, VISION:Workbench for DOS automatically offers you a new box for the next segment. Type ORDER in the new box (see Figure 3-14).

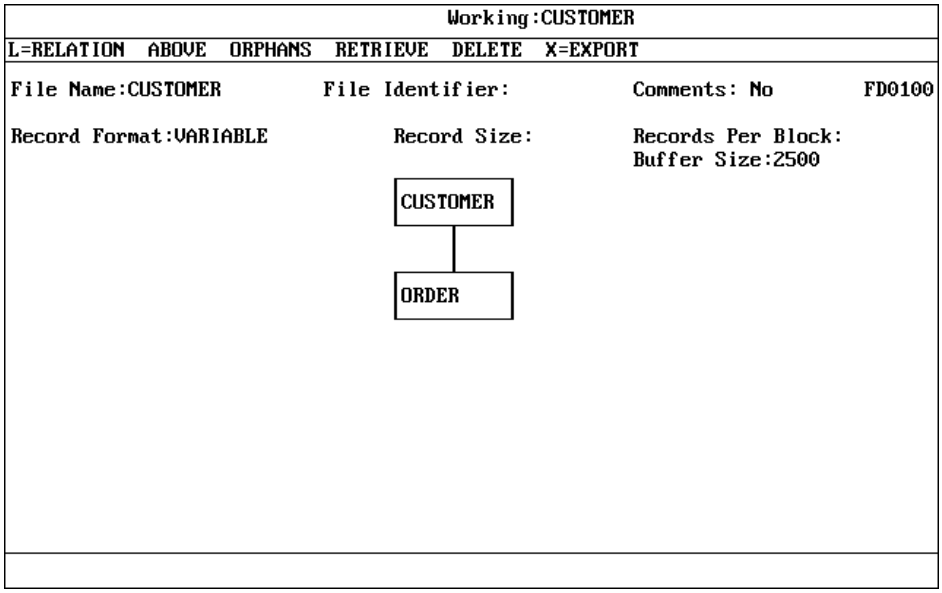



Figure 3-14 File Definition Window with Two Segments

Using simple cursor movements, you can create segments to the right, left, or below the ORDER segment. If you tab from a box, a new segment box is formed to the right. If you press Enter, a new box is formed below. If you press the Shift and

Tab keys simultaneously (called back-tab), a new segment box is formed to the left. To insert a new segment box above an already defined box, press F4 to toggle on the Above function and press F9 to insert the segment box.

Create a new box to the right of the ORDER segment. The new segment is called INSTALL (see Figure 3-15).

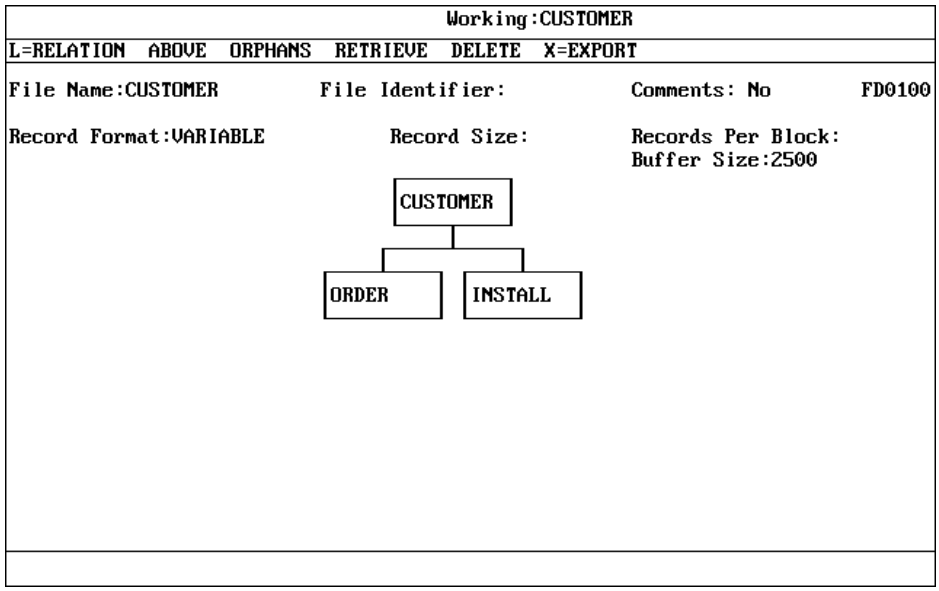



Figure 3-15 File Definition Window with Three Segments

Figure 3-16 and Figure 3-17 show the final steps in defining the structure of the CUSTOMER file.

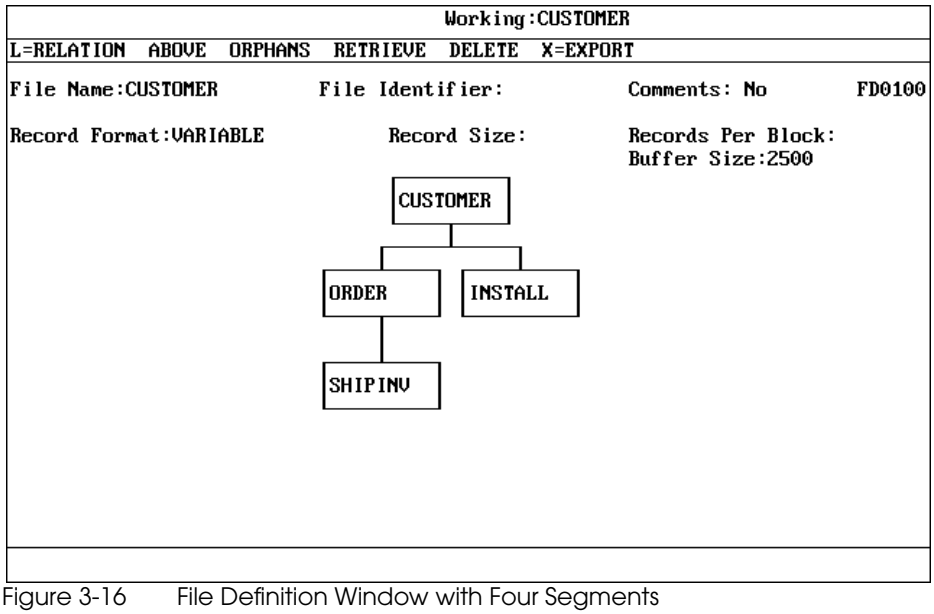



Figure 3-16 File Definition Window with Four Segments

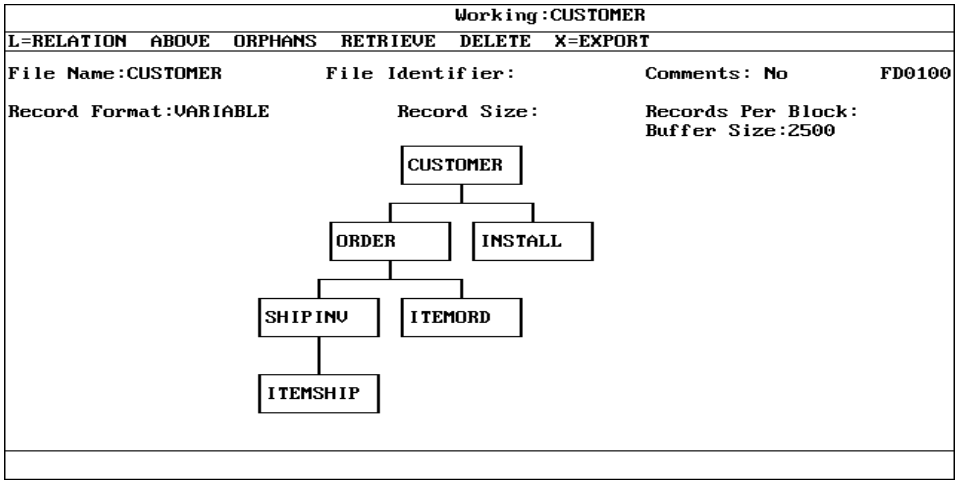



Figure 3-17 Complete File Definition Window

Once the file structure has been defined, highlight the segment you want to define (CUSTOMER) and press F8. The Segment Definition window for the CUSTOMER segment appears in Figure 3-18.

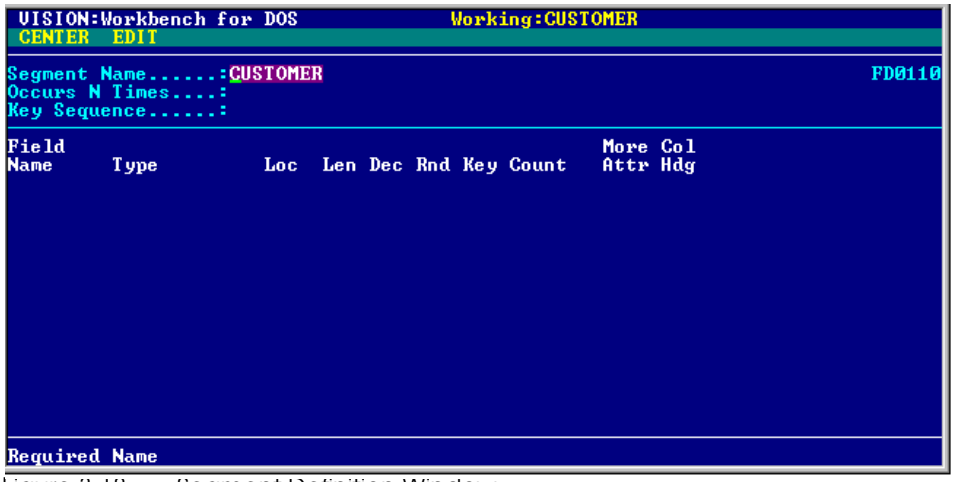



Figure 3-18 Segment Definition Window

Segment Name is automatically entered. You do not need to fill in the Occurs N Times field because this is the root. At the Key Sequence entry, the HELP windows in Figure 3-19 and Figure 3-20 explain your options.

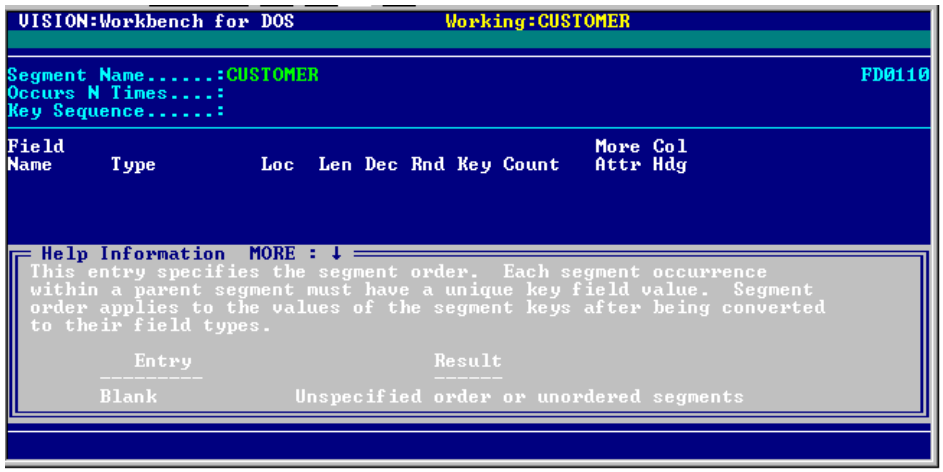



Figure 3-19 HELP Window for Key Sequence

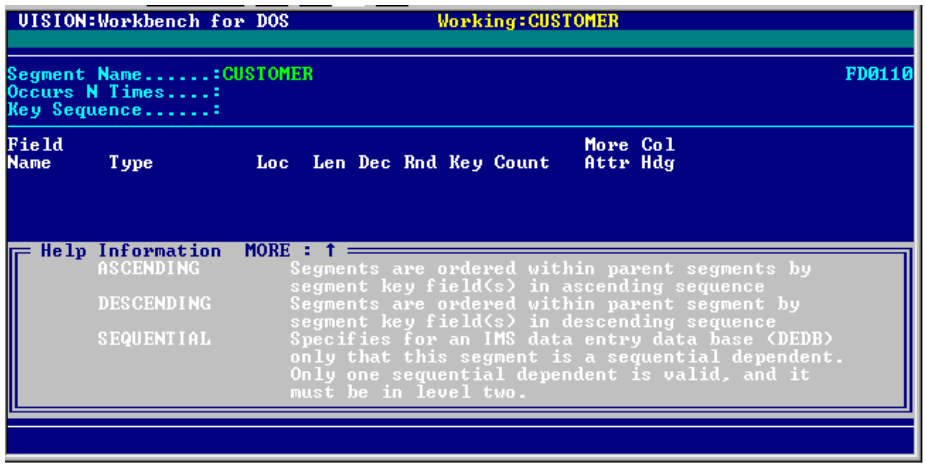



Figure 3-20 More HELP for Key Sequence

In this example, the key sequence is ascending (see Figure 3-21). Under Field Name, CUSTNO defines the first field of the customer segment. The Type field specifies that CUSTNO is a character field. The Location is 1 (default). The Length is 5. Remember that the HELP window explains your choices for each entry.

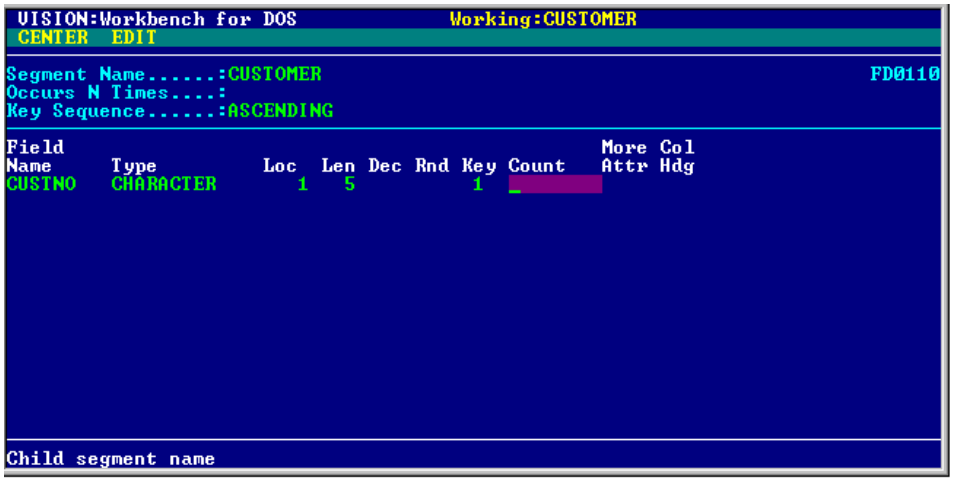



Figure 3-21 Define a Field

The Dec column lets you specify number of decimal places in a numerical field. Rnd lets you specify that the field should be rounded, not truncated, when used in an arithmetic expression.

The More Attr entry gives you the ability to specify output edit criteria in your file definition, so all of your applications can use them. There is no need for later procedural programming to edit your fields for output. Figure 3-22 displays the popup window for the More Attr entry.

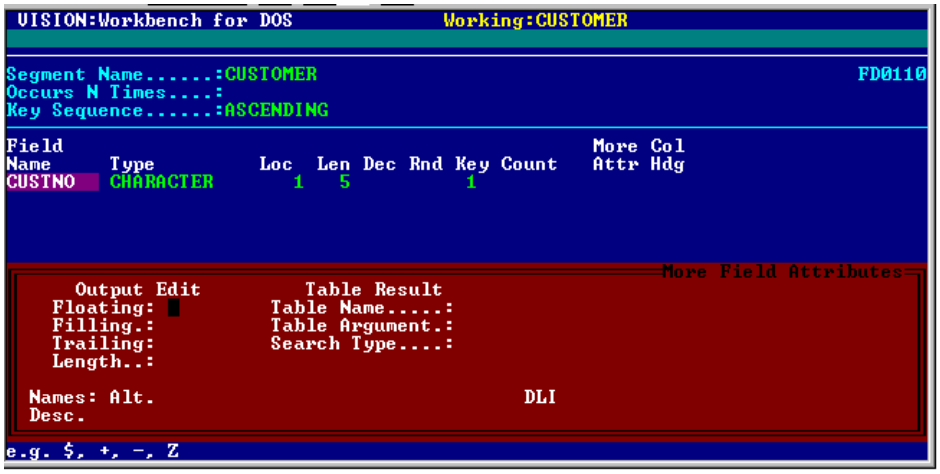



Figure 3-22 More Attr Window

Figure 3-23 through Figure 3-26 show you the HELP window explanations.

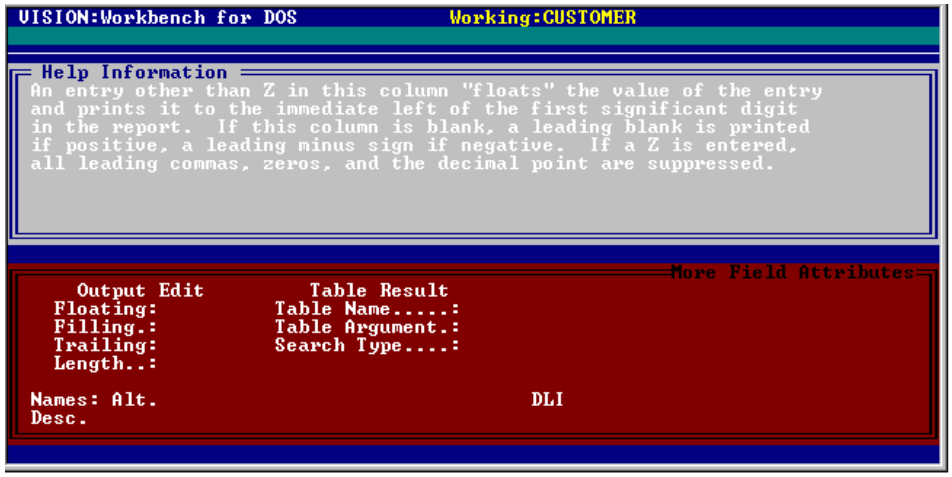



Figure 3-23 More Attr HELP Window for Floating

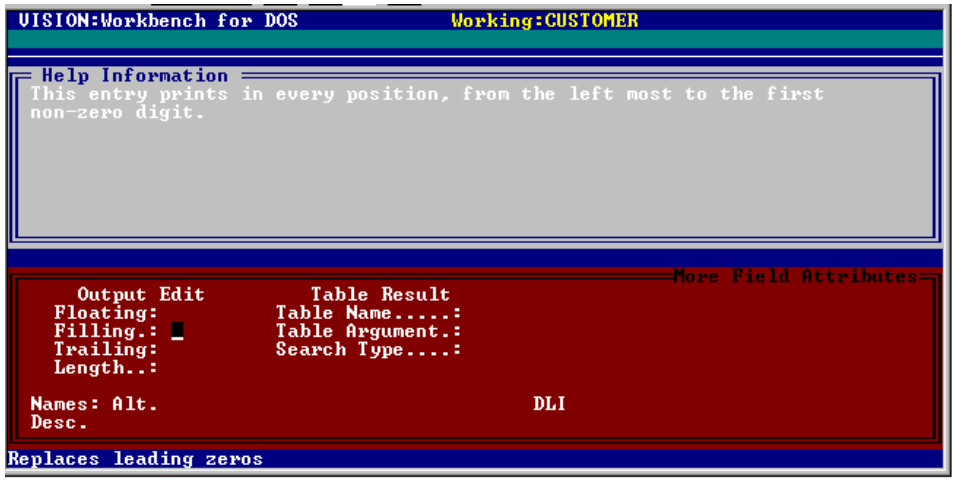



Figure 3-24 More Attr HELP Window for Filling

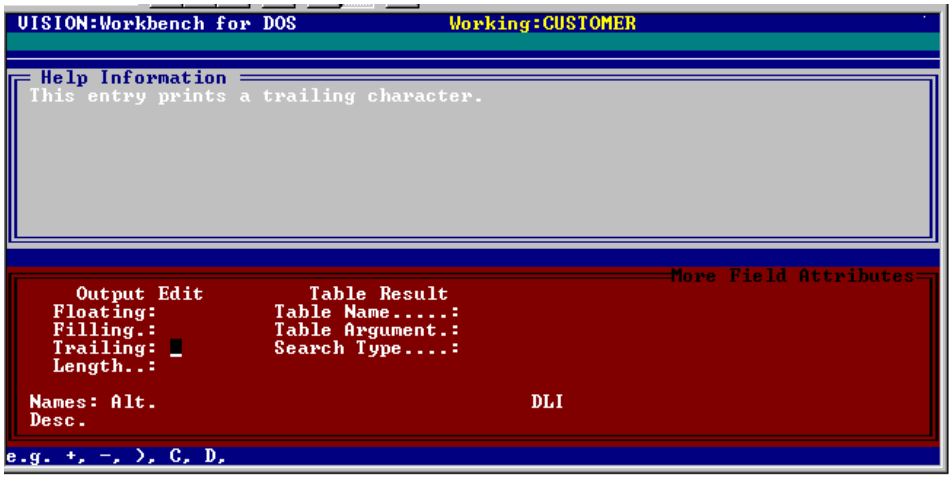



Figure 3-25 More Attr HELP Window for Trailing

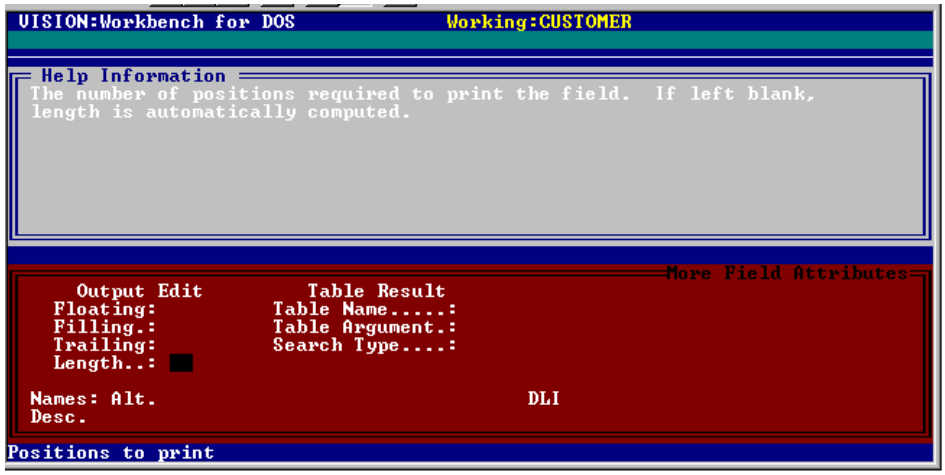



Figure 3-26 More Attr HELP Window for Length

The Col Head entry (column heading) specifies what column headings VISION:Builder should use in all future reports. You can leave this entry blank or you can override these column headings when defining a report in an application.

You can specify up to nine lines of heading. In this case, CUSTOMER NUMBER will be displayed on two lines, as shown in Figure 3-27. When column headings have been defined, the letter Y appears in the Col Hdg field.

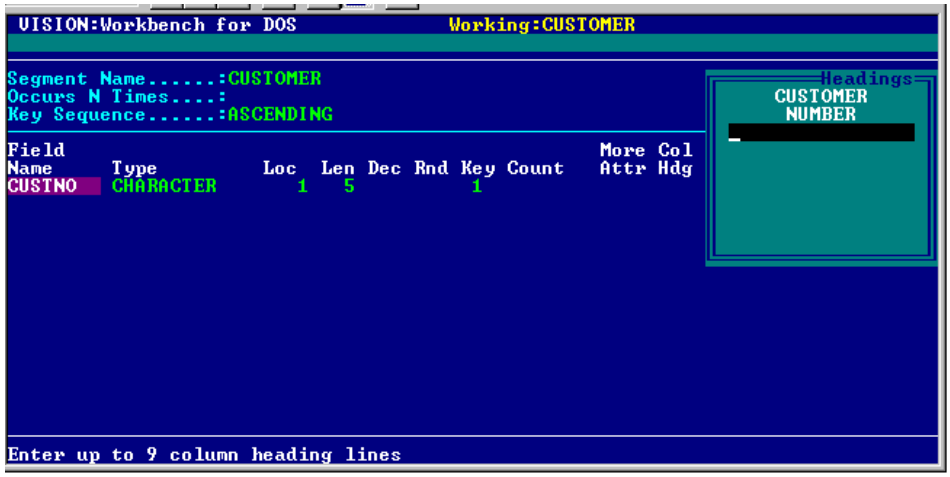



Figure 3-27 Enter a Column Heading for a Field

This process of defining segments is repeated for the rest of the Customer file. Once defined, the file definition is ready for export to the host.

For more information about exporting definitions to the host, see Appendix B, *VISION:Builder System Component Relationships*.

## Catalog the File Definition in the Common Library

Once defined and exported to the host, the file definition statements are used as input for a definition/maintenance run. Once cataloged in the common library, the definitions are automatically available to any application when the input file is specified on the Files Used window.

Generally, definition runs are done once, unless the definition changes. The file characteristics can be changed by a new definition/maintenance run, which catalogs the changes.

For more information about cataloging file definitions, see Chapter 19, *Cataloging*.

# Defining a VSAM File

You can access VSAM files just like any other file. The definition process for VSAM files is the same as for any other file you define, but there are some special considerations. These special considerations are described in this section.

You can define VSAM files to be used by VISION:Builder as input and output❹ master files, coordinated files, transaction files,❹ and subfiles.

## VSAM Record Structures

VSAM supports three types of file organizations, two of which are supported in VISION:Builder.

- KSDS (key sequenced) Supported in VISION:Builder
- ESDS (entry sequenced) Supported in VISION:Builder
- RRDS (relative record) Not supported in VISION:Builder

VSAM also supports alternate indexes. An alternate index provides an alternate entry to a KSDS or ESDS file. Alternate indexes are supported in VISION:Builder.

For KSDS files, the key field in your root segment must be in ascending order by key value and each key must be unique; only one key field is permitted. For ESDS files to be used with VISION:Builder, an arbitrary root key field must be defined even though ESDS files do not have a key.

By using an alternate key, ESDS files can be accessed sequentially as a master file or accessed directly as an indexed coordinated file. ESDS records can be read, updated, and inserted. When changes are made to the base cluster, both the prime and alternate indexes are automatically updated to reflect the changes.

## VSAM Cluster Definitions

To define a VSAM file to VISION:Builder, you need information from the VSAM cluster definition. Figure 3-28 shows a sample VSAM cluster definition for the VSAM CUSTOMER file. The cluster definition defines the characteristics of your VSAM file to the VSAM access method. Many of these same characteristics must also be defined in the file definition.

```
DEFINE CLUSTER -
    (NAME (PSO.CBT.CUSTOM) -
    FILE(CBTCUST) -
    RECORDS(25 5) -
    VOLUMES(DOSV01) -
    RECORDSIZE(1000 2000) -
    INDEXED KEYS(5,0)) -
    DATA -
      (NAME(PSO.CBT.CUSTOM.DATA) -
      CONTROLINTERVALSIZE(2048)) -
    INDEX -
      (NAME(PSO.CBT.CUSTOM.INDEX)) -
    CATALOG(AMASTCAT)
```

Figure 3-28 A Sample VSAM Cluster Definition the CUSTOMER File Definition

The FILE parameter is the name that identifies the VSAM file. The RECORDSIZE parameter in the cluster definition contains the average and maximum lengths of the data record. The maximum record length must be specified in the Buffer Size entry on the File Definition window. The maximum record length also influences the control interval specification in the cluster definition. Refer to IBM's Access Method Services manual for more detailed information about cluster definitions.

The cluster definition indicates the type of VSAM file. The INDEXED parameter indicates that the file is KSDS. The NONINDEXED parameter indicates that the file is ESDS. The NUMBERED parameter indicates that the file is RRDS. RRDS is not supported in VISION:Builder.

For INDEXED and NONINDEXED, the KEY parameter in the cluster definition describes the length and relative location of the record key. In the sample cluster definition shown in Figure 3-28, the record key is CUSTNO. It is located at the beginning of the record (relative location zero) and its length is 5 bytes. The file definition for a KSDS must match the key specifications in the cluster definition.

### VSAM Alternate Indexing

A VSAM alternate index is a VSAM feature that provides an entry to a key sequence data set (KSDS) or entry sequence (ESDS) data set through a field other than the primary root key. Through an alternate index path, you can process sequentially or directly.

You can use an alternate index path by cataloging a file definition with the alternate key field as the record key. You can also make the alternate key field the record key by temporarily overriding the file definition instream. Refer to the *VISION:Builder Reference Summary* for detailed information about overriding the file definition instream.

Once you define the alternate index to VSAM, VSAM automatically maintains the index whenever changes are made to the VSAM base cluster. When you want to process the VSAM file through the alternate index, the job control language (JCL) for your application must identify the alternate index path.

Another alternative is to process the alternate index as a file. When you retrieve data in the alternate index, you can access it sequentially or directly.

If you want to process the alternate index as a file, you must also create a file definition for the alternate index. In the alternate index file definition, specify the following:

- An A (for alternate index) in the Record Format entry on the File Definition window.
- The control interval size in the Buffer Size entry on the File Definition window. This is used in the same way the JCL buffer size is used for other types of files.

## Defining an IMS File

The VISION:Builder IMS interface provides transparent processing of databases managed by IBM's information management systems (IMS). The database management facilities of IMS are available in the following forms: IMS for OS/390 and DL/I for VSE. Throughout this discussion, the term IMS will be used to refer to all versions of the product, regardless of the environment. The term DL/I refers to the Data Language Interface used to communicate with IMS.

## IMS Record Structure

The structure of an IMS record is that of a hierarchical arrangement of related segments just as in VISION:Builder. Each record contains one root segment and up to 255 segment types. A dependent segment of the root can in turn be a parent of other dependent segments. The segments dependent on the root are at level two, the root at level one. Dependents of level two segments are at level three, and so forth. While VISION:Builder supports nine levels, an IMS structure can have up to 15 levels.

A segment can occur any number of times and space is never reserved for future occurrences. The order in which the segments are accessed by IMS depends on whether the segment is keyed or not. Uniquely and non-uniquely keyed segments are accessed in ascending key sequence with multiple occurrences of key values in the order in which they were inserted. Unkeyed segments are maintained in the order of insert. There is a RULE parameter in the database description (DBD) that determines where, in a chain of occurrences, a segment will be inserted. IMS also permits variable length segments.

The IMS database description (DBD) is a control block that defines a database. In a DBD, all segments are named, their length defined, and their position in the hierarchical structure established. The DBD also describes segment key fields and other fields on which IMS can select a segment occurrence. Physical attributes of the database such as the organization and access method, the physical record size, and blocking factor are also in the DBD. It should be noted that the record size and block size in the DBD are not related to the total size of an IMS record. Instead, they have to do with the amount of data that is transferred in a physical I/O.

The program specification block (PSB) is another IMS control block. The PSB describes the application program's view of the database(s). A PSB consists of one or more program communication blocks or PCBs. Each PCB describes one program's view of one database. A PCB consists of a set of statements that names the segments of a database to which the program has access or is sensitive.

Not all segments in the DBD need be named in the PCB, only those to which the program needs access. However, the segments named in the PCB must maintain the hierarchical structure of the DBD. Levels cannot be skipped; left to right sequence of segments cannot be altered.

The PCB also describes the manner in which the application program can access a segment (such as retrieval only, restricted update, or all update). The PCB can also be used to restrict the view of a segment to a set of fields. The application program sees only those fields in the I/O area instead of an entire segment. Complete details on the DBD and PSB can be found in IBM's IMS Utilities Reference Manual for the appropriate environment.

## VISION:Builder and IMS

In processing an IMS database, VISION:Builder uses the file definition to relate the structure to the one known to IMS. Because both VISION:Builder and IMS use similar hierarchical structures, relating the two becomes a matter of mapping the IMS structure in VISION:Builder terms.

Some of the characteristics of mapping the IMS structure are closely tied to the type of VISION:Builder processing. See IMS Processing on page 16-1 before defining the record to VISION:Builder. VISION:Builder supports all IMS access methods and organizations.

Because IMS makes the processing of logical structures transparent to the application program, this discussion also applies to logical structures. IBM's IMS Data Base Administration Guide and IMS Utilities Reference Manual discuss details on how logical databases are defined. VISION:Builder can also be used with the secondary indexing feature of IMS. See Secondary Indexing on page 3-31 for details.

In general, it is sufficient to map only the structure described in the PCB. The PCB must limit its view of the database to nine levels and 99 segment types. Each segment in the PCB must correspond to a VISION:Builder segment.

If field sensitivity is used, the VISION:Builder segment should consist of the sensitive fields concatenated rather than the entire segment. The choice of segment numbers is arbitrary, provided the IMS hierarchical structure is maintained. The VISION:Builder segment names must be the same as the IMS names because VISION:Builder uses that information to identify the segments. The VISION:Builder segment keys must also be the same as defined in the DBD. Optionally, non-key fields can be specified as search fields to VISION:Builder (an S in the Key entry on the Segment Definition window).

Search fields can be used by VISION:Builder in MOSAIC processing or preselection procedures. Memory optimized (MOSAIC) processing brings, at most, one occurrence of each segment in storage at any one time (standard processing brings the entire logical record into the VISION:Builder work areas). In MOSAIC processing, a segment is not retrieved until it is referenced, either explicitly or implicitly.

Both MOSAIC processing and preselection procedures result in VISION:Builder building qualified SSAs (segment search arguments). A preselection procedure causes a qualified, Boolean SSA to be generated from the information in the procedure. This SSA is used during the retrieval of a segment and, as such, acts as a filter on the segments returned to VISION:Builder. This can restrict access to particular segment occurrences or enhance performance by returning to VISION:Builder only those segment occurrences actually needed by the application for both MOSAIC and standard processing. For more information about MOSAIC processing and preselection procedures, see Chapter 16, *Database Access Methodology*.

Because IMS segments always occur a variable number of times, they must never be defined as fixed occurring to VISION:Builder. With IMS, VISION:Builder does not require a Count entry for the variable occurring segment. However, if defined, VISION:Builder will maintain them. You should be aware that this can result in unnecessary updates to a parent segment when the child is being updated.

IMS does not require that segments have a unique key as does VISION:Builder. For the standard VISION:Builder retrieval, these segments can have any field or combination of fields specified as keys. For update processing, either standard or

MOSAIC, segments must be defined with the U operand on the FIELD statement in the DBD. If more than one key field is defined to VISION:Builder, the secondary key fields must be IMS search fields.

The VISION:Builder segment ordering can also be applied with caution to IMS databases. VISION:Builder orders segments with numeric keys in algebraic sequence while IMS uses EBCDIC sequence. Unexpected results could occur if ordering were specified on these segments. For example, suppose a segment is defined in the DBD as having a one-byte numeric key. On the database, there are the following occurrences (in hexadecimal): 00, 0F, 70, 7F, 80, 8F, F0, and FF. These are in EBCDIC collating sequence. However, if the segment were defined as ascending, VISION:Builder would interpret these values as +0, +15, +112, +127, -128, -113, -16, and -1, which are not in algebraic order.

VISION:Builder does not require that segments be ordered for standard retrieval. Segments can be specified as having any order that accurately reflects the progression of segments. For update with standard processing, segment ordering is required. However, for MOSAIC retrieval, ordering is required unless only one hierarchical leg is defined in the PCB/FD (program communications block/file definition). See Chapter 16, *Database Access Methodology* for a discussion of standard and MOSAIC processing.

VISION:Builder always assumes that packed fields are signed. This could lead to unexpected results when IMS keys or search fields actually contain unsigned packed fields. Because IMS does a byte-by-byte comparison of the fields in an SSA, an F sign in the data will never compare equal to a C sign in the SSA.

When using standard processing, it is possible to define a file definition that describes all segments in the DBD instead of just those in the PCB (provided the DBD structure is within the VISION:Builder limits of nine levels and 99 segment types). Any reference to a field in a segment to which the PCB is not sensitive is treated like any reference to a non-existent segment. The only impact that this would have on processing is in the compiling phase. Larger file definitions require more memory. If MOSAIC processing is to be used, the file definition must map only those segments in the PCB.

IMS variable length segments can be processed by VISION:Builder. The segment should be defined as having the maximum length as defined in the DBD. The first 4 bytes of the data portion of the segment will contain the length. It is your responsibility to process only the appropriate fields of the segment.

For OS/390 users, IMS batch message processing (BMP) applications can be checkpointed and restarted. Refer to the *VISION:Builder for OS/390 Environment Guide* for details.

## Write a DBD/PCB from a VISION:Builder File Definition

The following example assumes that a VISION:Builder file definition exists and that an IMS DBD and PCB have to be written to coincide with it. Figure 3-29 is a diagram of an existing VISION:Builder file definition and Figure 3-30 shows headings from the glossary of this file definition. By including segment names, key field names, and key field lengths, the task of writing the IMS control blocks is

made much easier. There is enough information in these portions of the listings to enable you to write the PCB and DBD with the exception of the key field start location.

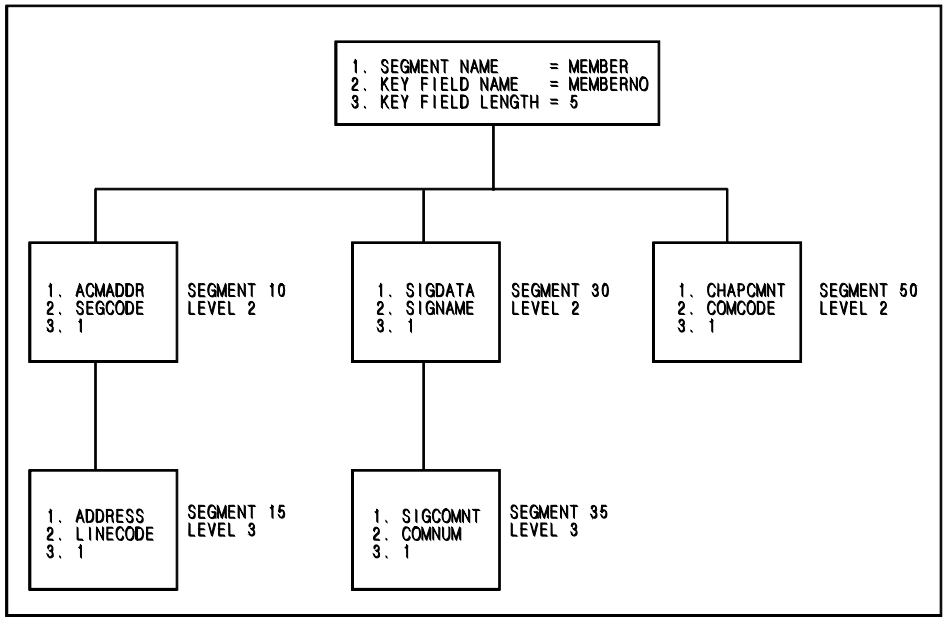



Figure 3-29 Diagram of a VISION:Builder File Definition

```
JAN 15, 1995 19.18.36
                                        DETAILED GLOSSARY BY LOCATION FOR                                GLOSSARY
                                          FILE DEFINITION - ACM-FILE
DATE CATALOGED -                                                                DATE LAST UPDATED -
CATALOGED BY -                                                                  LAST UPDATED BY -
EXPIRATION DATE -                                                               DATE LAST USED -
FILE IDENTIFICATION = ACMFILE                                                   RECORD FORMAT = DL/I
NUMBER OF SEGMENTS IN FILE = 6
NUMBER OF FIELDS IN FILE = 19                                                   BLOCK SIZE = 3500

JAN 15, 1995 19.18.36
                                        DETAILED GLOSSARY BY LOCATION FOR                                GLOSSARY
                                          FILE DEFINITION - ACM-FILE

                                       ********************************
                                       *     SEGMENT 10, LEVEL 2      *
                                       *   SEGMENT NAME = ACMADDR     *
                                       ********************************
COUNT FIELD FOR SEGMENT = ADDR-CTR    SEGMENT ORDER =  ASCENDING     KEY FIELD 1 = SEGCODE    TYPE = C   LENGTH = 1
SEGMENT SIZE = 7
NUMBER OF FIELDS IN SEGMENT = 3

JAN 15, 1995 19.18.36
                                        DETAILED GLOSSARY BY LOCATION FOR                                GLOSSARY
                                          FILE DEFINITION - ACM-FILE

                                       ********************************
                                       *     SEGMENT 15, LEVEL 3      *
                                       *   SEGMENT NAME = ADDRESS     *
                                       ********************************
COUNT FIELD FOR SEGMENT = LINE-CTR    SEGMENT ORDER =  ASCENDING     KEY FIELD 1 = LINECODE   TYPE = C   LENGTH = 1
SEGMENT SIZE = 60
NUMBER OF FIELDS IN SEGMENT = 4
```

Figure 3-30 VISION:Builder Glossary Headings (Page 1 of 2)

```
JAN 15, 1995 19.18.36
                                DETAILED GLOSSARY BY LOCATION FOR                              GLOSSARY
                                  FILE DEFINITION - ACM-FILE

                                *********************************
                                *     SEGMENT 30, LEVEL 2       *
                                *   SEGMENT NAME = SIGDATA      *
                                *********************************
COUNT FIELD FOR SEGMENT = SIG-CTR     SEGMENT ORDER =  ASCENDING     KEY FIELD 1 = SIGNAME    TYPE = C   LENGTH = 1
SEGMENT SIZE = 30
NUMBER OF FIELDS IN SEGMENT = 3

JAN 15, 1995 19.18.36
                                DETAILED GLOSSARY BY LOCATION FOR                              GLOSSARY
                                  FILE DEFINITION - ACM-FILE

                                *********************************
                                *     SEGMENT 35, LEVEL 3       *
                                *   SEGMENT NAME = SIGCOMNT     *
                                *********************************
COUNT FIELD FOR SEGMENT = S-COMCTR    SEGMENT ORDER =  ASCENDING     KEY FIELD 1 = COMNUM     TYPE = C   LENGTH = 1
SEGMENT SIZE = 64
NUMBER OF FIELDS IN SEGMENT = 2


JAN 15, 1995 19.18.36
                                DETAILED GLOSSARY BY LOCATION FOR                              GLOSSARY
                                  FILE DEFINITION - ACM-FILE

                                *********************************
                                *     SEGMENT 50, LEVEL 2       *
                                *   SEGMENT NAME = CHAPCMNT     *
                                *********************************
COUNT FIELD FOR SEGMENT = COM-CTR     SEGMENT ORDER =  ASCENDING     KEY FIELD 1 = COMCODE    TYPE = C   LENGTH = 1
SEGMENT SIZE = 64
NUMBER OF FIELDS IN SEGMENT = 2
```

Figure 3-30 VISION:Builder Glossary Headings (Page 2 of 2)

There are specifications for the DBD and PCB that are not associated with the VISION:Builder file definition. These are noted briefly, but because they are beyond the scope of VISION:Builder, they are not included in the explanation.

Figure 3-31 shows the source listing of a PCB developed from the diagram and the glossaries of the file definition. The first statement of the PCB defines the characteristics of this PCB. The TYPE and PROCOPT keywords require knowledge of IMS and, in this case, specify that the PCB is for a database application and allow you to retrieve, insert, replace, and delete segments in the database.

The DBDNAME keyword matches the value provided on page 1 of the glossary under file identification (see Figure 3-30). It also agrees with the DBD itself, which is described in Write a DBD/PCB from a VISION:Builder File Definition on page 3-25.

The KEYLEN keyword can be easily calculated from the diagram. In Figure 3-29, start at the level 1 segment and proceed down each possible path and add the key length values together. For example, the path from segment 1 through segment 10

and segment 15 results in a value of 5 + 1 + 1 = 7. Segment 1 through segments 30 and 35 is again 5 + 1 + 1 = 7. The path from segment 1 through segment 50 is 5 + 1 = 6. The maximum value is 7, which is the value used in the KEYLEN keyword.

```
                                                                                                    PAGE   3

                                                                                          ASM M V 05 15.12
 LOC OBJECT CODE  ADDR1 ADDR2 STMT  SOURCE STATEMENT
                                   3        PCB  TYPE=DB,DBDNAME=ACMFILE,PROCOPT=AP,KEYLEN=7,POS=M
                                   4        SENSEG NAME=MEMBER,PARENT=0
                                   5        SENSEG NAME=ACMADDR,PARENT=MEMBER
                                   6        SENSEG NAME=ADDRESS,PARENT=ACMADDR
                                   7        SENSEG NAME=SIGDATA,PARENT=MEMBER
                                   8        SENSEG NAME=SIGCOMNT,PARENT=SIGDATA
                                   9        SENSEG NAME=CHAPCMNT,PARENT=MEMBER
                                  10        PSBGEN LANG=ASSEM,PSBNAME=ACMRUN
                                 103        END
```

Figure 3-31 PCB Source Listing

The next statements needed for the PCB are the SENSEG statements, which can also be developed from the diagram. The NAME keyword is filled from the name associated with each segment in the file definition. Their parent can be determined by starting at the bottom of each branch and filling in the name of the next segment up the line. Because the level 1 segment has no parent, its entry is 0.

The last statement is the PSBGEN statement. The LANG keyword specifies ASSEM although COBOL could also have been used. This entry will always be the same for a PCB to be used with VISION:Builder. The PSBNAME keyword can be arbitrarily selected by you to agree with IMS naming conventions and used in the JCL when the application is executed.

Figure 3-32 shows a DBD for a DL/I HIDAM database and Figure 3-33 contains lines extracted from the file definition glossary for each of the key fields. Because many of the keywords on the DBD are strictly IMS in nature, this part of the explanation will point only to those entries that come from the file definition.

```
                                                                                                    PAGE   2

                                                                                          ASM M V 05 15.11
 LOC OBJECT CODE  ADDR1 ADDR2 STMT  SOURCE STATEMENT
                                   1        DBD   NAME=ACMFILE,ACCESS=HIDAM
                                   2 LEVEL1  DATASET  DD1=ACMHIDAM,DEVICE=3330,MODEL=1,BLOCK=3500,SCAN=2
                                   3+*,           3330 DISK STORAGE
                                   4+*,           MODEL 1
                                   5        SEGM  NAME=MEMBER,PARENT=0,BYTES=80,FREQ=1000,POINTER=TWINBWD
                                   6        LCHILD  NAME=(INDEX,ACMINDEX),PTR=INDX
                                   7        FIELD NAME=(MEMBERNO,SEQ,U),BYTES=5,START=2,TYPE=C
                                   8 LEVEL2  DATASET  DD1=ACMADSEG,DEVICE=3330,MODEL=1,BLOCK=1000,SCAN=5
                                   9+*,           3330 DISK STORAGE
                                  10+*,           MODEL 1
                                  11        SEGM  NAME=ACMADDR,PARENT=((MEMBER,DBLE)),BYTES=30,PTR=TWINBWD
                                  12        FIELD NAME=(SEGCODE,SEQ,U),BYTES=1,START=1
                                  13 LEVEL1  DATASET
                                  14        SEGM  NAME=ADDRESS,PARENT=ACMADDR,BYTES=60
                                  15        FIELD NAME=(LINECODE,SEQ,M),BYTES=1,START=1
                                  16        FIELD NAME=LINETYPE,BYTES=1,START=2
                                  17        FIELD NAME=LINENUMB,BYTES=3,START=3
                                  18 LEVEL2  DATASET
                                  19        SEGM  NAME=SIGDATA,PARENT=((MEMBER,DBLE)),BYTES=30,PRT=HIERBWD
                                  20        FIELD NAME=(SIGNAME,SEQ,U),BYTES=1,START=1
                                  21        SEGM  NAME=SIGCOMNT,PARENT=SIGDATA,BYTES=64,PTR=HIERBWD
                                  22        FIELD NAME=(COMNUM,SEQ,U),BYTES=1,START=1
                                  23        SEGM  NAME=CHAPCMNT,PARENT=MEMBER,BYTES=64
                                  24        FIELD NAME=(COMCODE,SEQ,U),BYTES=1,START=1
                                  25        DBDGEN
```

Figure 3-32 Sample DBD

The NAME keyword of the DBD statement comes from the file identification listed on page 1 of the glossary. The BLOCK keyword of the LEVEL 1 DATASET statement is the block size from page 1 of the glossary. None of the other keywords comes from the VISION:Builder file definition; however, the DD1 keyword does specify which DDNAMEs will be required in the JCL when the job is run.

The SEGM statement has three keywords that come from the glossary. The NAME and PARENT values are selected exactly as described for the PCB. The BYTES keyword entry can be entered from the segment size on the glossary. By comparing Figure 3-32 to Figure 3-30 on page 3-26, you should be able to see the correspondence for that entry.

The LCHILD statement has meaning to IMS and cannot be derived from the file definition glossary, but there is data for specifying the FIELD statement: the first subfield of the NAME keyword is the field name of key field 1; the BYTES keyword is the length; and the TYPE keyword comes from type on the same line of the listing. The only other keyword on this statement is START, which comes from the body of the glossary where it is the field location value for the field named as key field 1. A comparison of Figure 3-30 and Figure 3-33 will show the relationship. Note that all key fields are flagged as search fields, which means that they must correspond to identically named fields in the DBD. Other fields which are not keys that have been defined as search fields are similarly flagged in the glossary listing.

```
MEMBERNO C/S   2   5                5   7
                                        1  ***  CHAPTER   ***
                                        2  ***  MEMBER    ***
                                        3  ***  NUMBER    ***


SEGCODE  C/S   1   1                1   1


LINECODE C/S   1   1                1   4
                                        1  ***  LINE      ***
                                        2  ***  CODE      ***


SIGNAME  C/S   1   1                1   1


COMNUM   C/S   1   1                1   1


COMCODE  C/S   1   1                1   1
```

Figure 3-33 Key Field Attributes

## Write a VISION:Builder File Definition from a DBD/PCB

Writing a VISION:Builder file definition from a DBD and PCB is similar to the process just described. First, specific entries are made on the File Definition window that result in a skeletal definition. The definition is completed based on the structure of the database segment. This structure must come from something similar to a record layout, because that information is not necessarily included in IMS control blocks.

Required VISION:Builder entries in this example depend on the actual definition of the data. The first entries are made on the File Definition window (see Figure 3-34). The File Identifier entry comes from the DBDNAME keyword on the PCB statement (see Figure 3-31 on page 3-28) and because this is a database file, the

Record Format entry must be DL/I DATABASE. The Buffer Size entry must be based on an estimate of the maximum logical record size. This estimate requires some knowledge of the data.

Next, the segments must be entered, coinciding with each SENSEG statement in the PCB. The field name entry in each segment box is the value from the NAME keyword of the SENSEG statement.

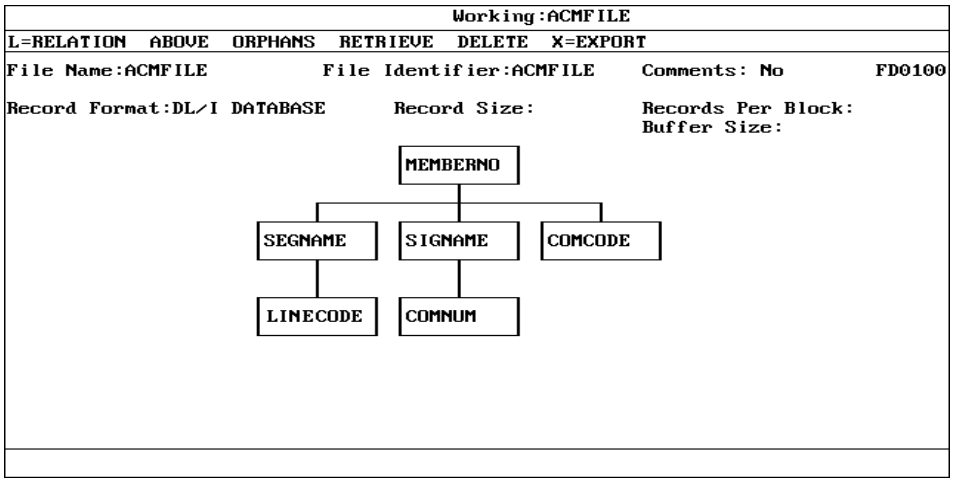



Figure 3-34 File Definition Window Using DBD / PCB

Figure 3-35 shows the Segment Definition window for the MEMBER segment. The Field Name entry is obtained from the NAME keyword of the FIELD statement (for example, MEMBERNO in line 7 of Figure 3-32 on page 3-28). The Key entry must be either a number, if the field is to be a VISION:Builder key, or the letter S, if it is a search field but not a VISION:Builder key.

The values for Location, Length, and Type are taken from START, BYTES, and TYPE, respectively. Note here that VISION:Builder requires that all field names within a file definition be unique; whereas, IMS only requires uniqueness within a segment. In case of conflicts, an alias (DL/I Alias entry on the Segment Definition window) can be provided to VISION:Builder for key and search fields. This alias should match the corresponding name in the DBD as this is now the field name VISION:Builder will use in generating SSAs.

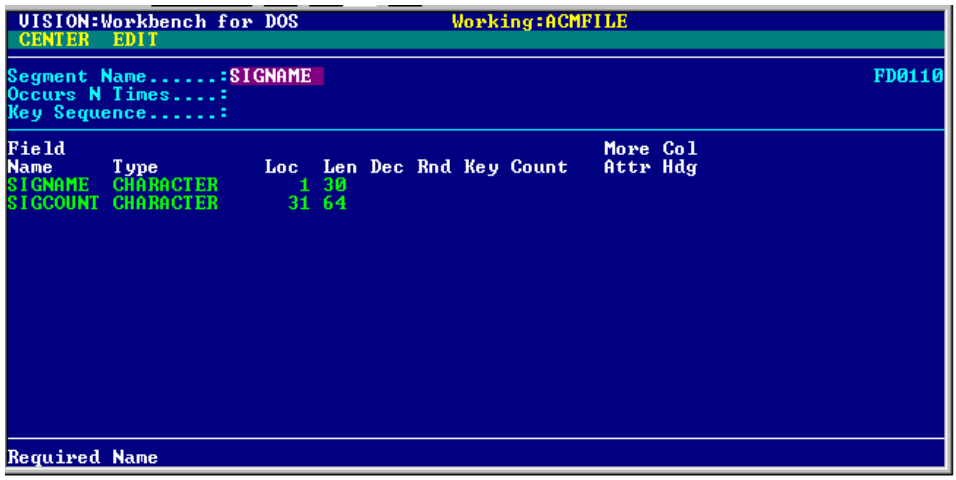



Figure 3-35 Segment Definition Window Using DBD / PCB

To complete the skeletal definition, the last byte in each segment must be defined. Field name entries for each segment must be supplied by you and are subject only to VISION:Builder naming conventions. If the last byte of the segment has not been previously defined as part of a field, you can arbitrarily define it as a one-byte field. The field length entry for each field is 1 and the field location entry is the value in the BYTES keyword of the SEGM statement for each segment.

## Secondary Indexing

Secondary indexing is a feature of IMS. A secondary index provides an entry point to a database using a field other than the primary root key. The segment containing the entry point field is the target segment and the segment containing the index key field(s) is the source segment. The source segment must be either the same segment as the target segment or a dependent of it. The IMS rule is that the target segment can never be a dependent of the source segment.

All secondary index structures can be processed by VISION:Builder. However, some care is required in defining the structure. If the target segment is the primary root, the primary structure is identical to the secondary index structure. In this case, the file definition used for the primary sequence can also be used for the secondary sequence if the primary sequence is bypassed as follows:

- Process the database as unsequenced with U in the Chk (sequence check) entry on the Files Used window (see Figure 3-36).
- Do not access the database randomly as M4OLD or as an indexed coordinated file.

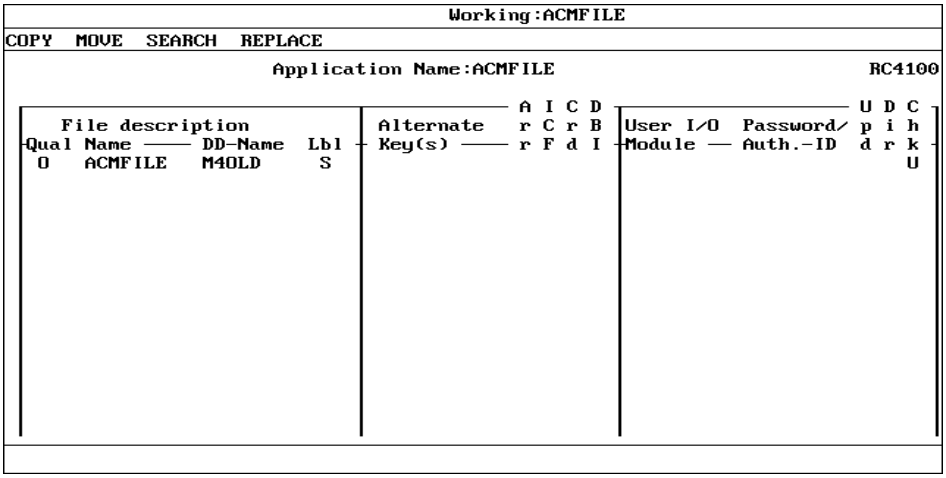



Figure 3-36 Files Used Window, Sequence Check

These restrictions do not apply if a new file definition describing the new root key is written for the secondary structure.

For all other cases, there must be a file definition that maps the secondary index structure. The PCB for the secondary index structure describes the segment hierarchy and sensitivity. If the target segment is a dependent segment, the PCB also illustrates the inversion of the segment hierarchy. Any segments not in the same leg of the structure as the source segment are not accessible.

The target segment becomes the root and the segments on which the target is dependent in the primary structure become the leftmost hierarchical leg, in inverted order. Figure 3-37 illustrates this.

**Note:** In Figure 3-37, in the Secondary Index Structure, note that there is an inverted structure and a loss of the F segment.

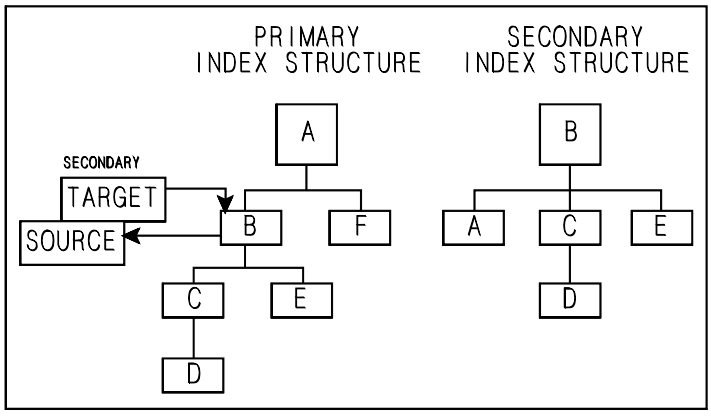



Figure 3-37 Inverted Secondary Index Structure

The field listed in the SRCH keyword of the XDFLD statement in the DBD becomes the root key. If the field is in the target segment, define that field as the new key. This occurs when the source and target are the same segment. If the source and target are not the same segment or if the field consists of non-contiguous fields, a virtual key field must be defined in the new root segment. The virtual key field is defined as the root segment key and will contain the value of the fields specified by the SRCH keyword. See VISION:Builder File Definitions and Secondary Indexing on page 3-33 for a description of the use of virtual key fields in file definitions.

It should be noted that with the exception of the segments on which the target segment is dependent, the remaining segment structure of the secondary structure is the same as that of the primary structure. Segments retain the same name, length, key fields, search fields, and hierarchical relationships. VISION:Builder is able to perform retrieval and transaction processing, both standard and MOSAIC, with little difficulty on secondary indexed databases. For more information about standard and MOSAIC processing, see Chapter 16, *Database Access Methodology*.

The update processing must abide by the IMS restrictions that prohibit changing the segment key fields of or performing create, insert, and delete actions on the target segment or any segment on which it is dependent in the primary structure. In the secondary structure shown in Figure 3-37, create, insert, and delete actions cannot be performed on segments A and B.

Preselection requests can be used on any dependent segment of a secondary structure. By using the virtual key field in the secondary structure file definition, they can also be used on root segments when the target and source are different segments and the fields specified by the SRCH keyword of the XDFLD statement are non-contiguous.

Secondary index structures can also be accessed both directly and sequentially as user-read coordinated files. Combining these two techniques in the same application run is an effective method of reading any subset of a database (see Chapter 6, *Input File Processing*).

## VISION:Builder File Definitions and Secondary Indexing

In general, a different VISION:Builder file definition is required for secondary indexed processing than the one used for normal (primary indexed) processing. There are two reasons why. First, the new file definition describes the secondary data structure. Second, the field specified by the NAME keyword of the XDFLD statement from the DBD through which secondary indexed access occurs must be defined as a virtual key field in the secondary root segment. The original segment key specifications (the Key entry on the Segment Definition window) must be blank.

A virtual key field is defined with keywords from the XDFLD statement for the secondary structure. This key field is available as though it is part of the root segment for all VISION:Builder processing and allows VISION:Builder to automatically process a DL/I segment that does not contain its secondary index

key field. A virtual key is treated just as any other database segment key by VISION:Builder. During DL/I calls, the virtual key cannot be overdefined or included in the segment; therefore, it does not appear in the DL/I file on disk.

The following rules must be followed when specifying a virtual key field on the Segment Definition window:

- The Field Name of the virtual key field must correspond to the name specified by the NAME keyword of the XDFLD statement in the DBD.
- The Location must be blank.
- The Length is the combined length of all fields specified by the SRCH keyword of the XDFLD statement in the DBD.
- The Type must be character.
- The Key entry must be V. No other key fields, except search fields, can be named for this segment. V is valid only for a root segment.

The example in Figure 3-38 and Figure 3-39 show the file definition for a two-level DL/I database accessed using a secondary index (declared using the PROCSEQ = keyword in the PCB).

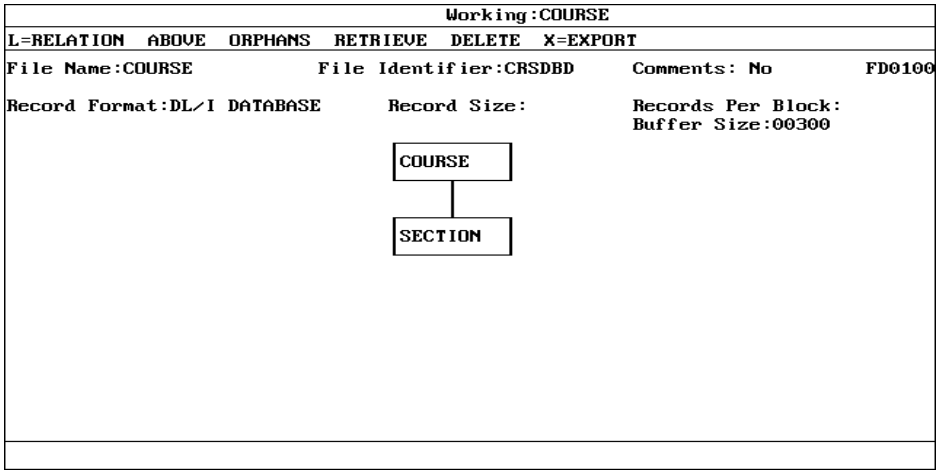



Figure 3-38 File Definition Window Using Virtual Key Field for Secondary Index

The example shows that the COURSE segment is accessed using the virtual key field CRSREFNO, a typical example in which the source segment is different from the target segment.

The virtual key field, CRSREFNO, must be defined using an XDFLD statement in the CRSDBD DBD. The key of the secondary index database is built from the field SCTREFNO in the second level segment (SECTION). The XDFLD statement must define SCTREFNO as the source segment search field for the secondary index. This relationship is transparent to the VISION:Builder file definition.

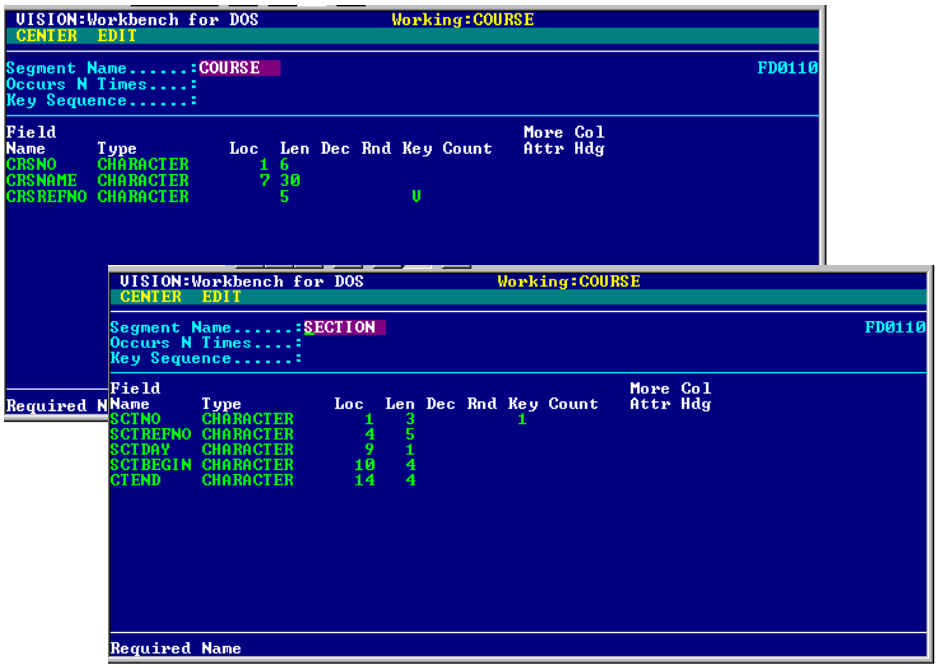



Figure 3-39 Segment Definition Windows Using Virtual Key Field for Secondary Index

The following examples describe the various combinations of target segment, source segment, and fields specified by the SRCH keyword of the XDFLD statement that can be expected when using secondary index processing.

## Example 1

- Target field is in the primary root segment.
- Source fields are in the primary root or a dependent segment.
- Single, contiguous, or non-contiguous fields specified by the SRCH keyword of the XDFLD statement.

Figure 3-40 shows both index structures.

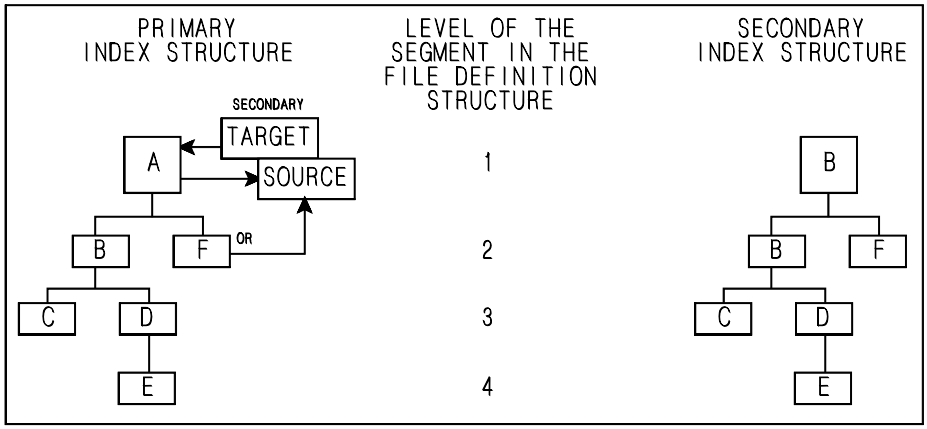



Figure 3-40 Example 1 Primary and Secondary Index Structures

### File Definition:

- In this example, the primary and secondary structures are the same. You can avoid making a file definition for the secondary structure by changing the following VISION:Builder statements.
  - If the database is on M4OLD, specify a U (unsequenced) in the Chk (sequence check) entry on the Files Used window and specify blank (sequential) in the Dir (direct-read) entry. This reads the file sequentially by the secondary sequence.
  - If the database is to be an indexed coordinated file, it cannot be used without a new file definition for the secondary structure. The new file definition must have a virtual key field as the record key.

### VISION:Builder Processing Allowed:

- Sequential read only.

### IMS Restrictions:

- Segment key fields must not be changed.

### Example 2

- Target field is in the primary root segment.
- Source fields are in the primary root or are dependent segments.
- Single, contiguous, or non-contiguous fields specified by the SRCH keyword of the XDFLD statement.

Figure 3-41 shows both index structures.

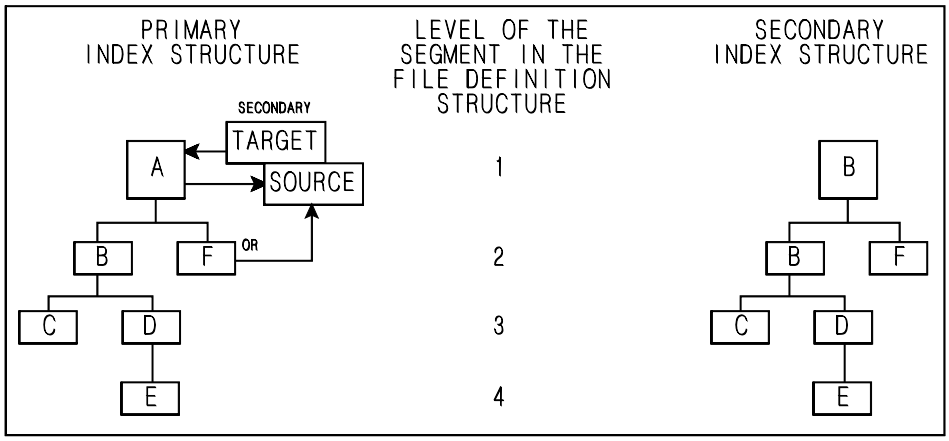



Figure 3-41 Example 2 Primary and Secondary Index Structures

### File Definition:

- In this example, as in Example 1, the primary and secondary structures are the same. If the full range of VISION:Builder processing is wanted, do the following:
  - Copy the primary index structure file definition.
  - Change the existing record key to a non-key field.
  - Make a virtual key field from the fields specified by the SRCH keyword of the XDFLD statement for the record key.

### All VISION:Builder Processing Allowed:

- Sequential and random retrieval on M4OLD.
- Start search.
- Update-in-place.❹
- ICF and sequential coordination.
- Transaction processing.❹
- Preselection on all segments.
- Full MOSAIC processing.

### IMS Restrictions:

- None.

### Example 3

- Target field is in any primary dependent segment.
- Source fields are in the same segment.
- Single, contiguous, or non-contiguous fields specified by the SRCH keyword of the XDFLD statement.

Figure 3-42 shows both index structures.

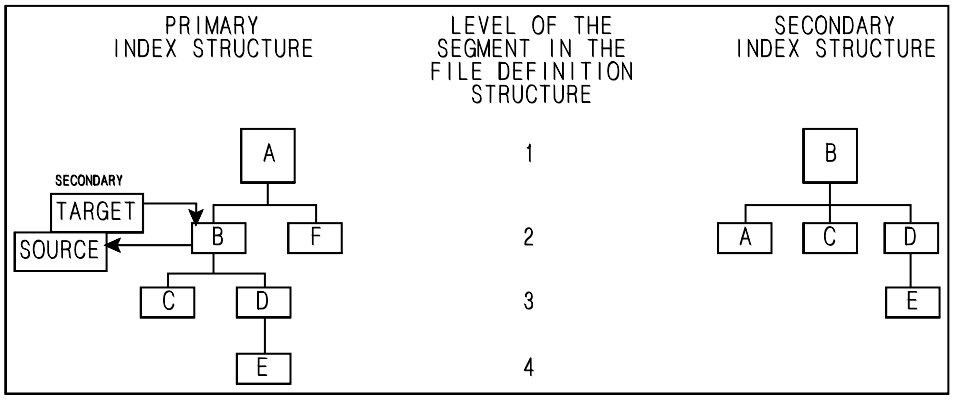



Figure 3-42 Example 3 Primary and Secondary Index Structures

### File Definition:

- Write a secondary index structure file definition.
- Make a virtual key field from the fields specified by the SRCH keyword of the XDFLD statement for the record key.
- The structure in the File Definition window should match the inverted structure exactly.

### All VISION:Builder Processing Allowed:

- Sequential and random retrieval on M4OLD.
- Start search.
- Update-in-place.❹
- ICF and sequential coordination.
- Transaction processing.❹
- Preselection on all segments.
- Full MOSAIC processing.

### IMS Restrictions:

- Create, insert, and delete are not allowed for target segment or any of its parents.
- Segment key fields of these segments must not be changed.

### Example 4

- Target field is in any primary dependent segment.
- Source fields are in a segment dependent on the target.
- Single, contiguous, or non-contiguous fields specified by the SRCH keyword of the XDFLD statement.

Figure 3-43 shows two possible index structures.

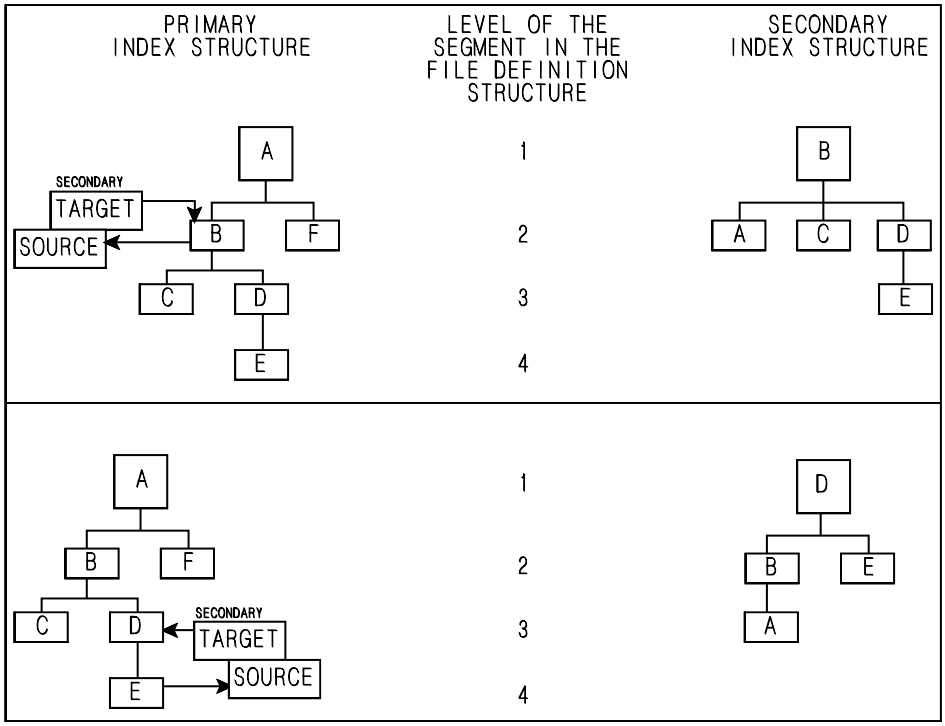



Figure 3-43 Example 4 Primary and Secondary Index Structures

### File Definition:

- Write a secondary index structure file definition.
- Make a virtual key field from the fields specified by the SRCH keyword of the XDFLD statement for the record key.
- The structure in the File Definition window should match the inverted structure exactly.

### All VISION:Builder Processing Allowed:

- Sequential and random retrieval on M4OLD.
- Start search.
- Update-in-place.❹
- ICF and sequential coordination.
- Transaction processing.❹
- Preselection on all segments.
- Full MOSAIC processing.

### IMS Restrictions:

- Create, insert, and delete are not allowed for target segment or any of its parents.
- Segment key fields of these segments must not be changed.

## Preparing a Secondary Structure Definition

The following steps are a guideline for preparing a file definition for a secondary structure.

1. Examine a listing of the PCB to be used. If the PCB statement includes a PROCSEQ keyword, secondary indexing is assumed for the database and an appropriate file definition should be prepared.
2. A PROCSEQ keyword indicates a secondary data structure. The PCB (not the DBD) reflects the hierarchical structure of the database and the new file definition should map this structure exactly.
3. Segment names in the file definition should be copied from the PCB SENSEG statements.
4. Key field names, locations, lengths, and field types can be determined in the usual manner for all segment types except the root of the secondary data structure.
5. Non-key fields can be defined in the usual manner.
6. When a PROCSEQ keyword is specified, the root segment of the secondary data structure is always accessed through a secondary index. The record key field described in the VISION:Builder file definition must be the field specified by the XDFLD statement. The characteristics of this index field can be determined in the following manner:
   - Examine a listing of the DBD named in the D13DNAME or NAME keyword of the PCB statement. Locate the SEGM statement that names the root segment of the secondary data structure. This SEGM statement describes the index target segment type for the structure and will be followed by at least one LCHILD statement before any other SEGM statements appear in the listing.

– From among these LCHILD statements, locate the one that names the secondary index for the structure. The secondary index name will be the second operand of the NAME keyword of the LCHILD statement. For example:

```
LCHILD          NAME = (PARTPTR,PARTINDX). . . .
                              secondary index name
```

PARTINDX will be the name specified by the PCB PROCSEQ keyword. Next, locate the first XDFLD statement following this LCHILD statement.

– The NAME keyword of the XDFLD statement specifies the index name to be used as the record key in the VISION:Builder file definition.

– The starting location of the record key can be found by examining the SEGMENT and SRCH keywords of the XDFLD statement:

If the name specified by the SEGMENT keyword differs from the segment name in the preceding SEGM statement, the index source segment is a dependent of the index target segment. The secondary index comes from the index source segment.

A virtual key field named for the secondary index must be defined for the secondary root segment. If the SEGMENT keyword is omitted or if it specifies the same segment name as in the preceding SEGM statement, the index source and index target segments are identical. The secondary index comes from the index source segment. The record key start location can be found by noting the leftmost (or only) field name specified by the SRCH keyword and finding that field's START keyword value among the FIELD statements for the segment. See Figure 3-44 for an example.

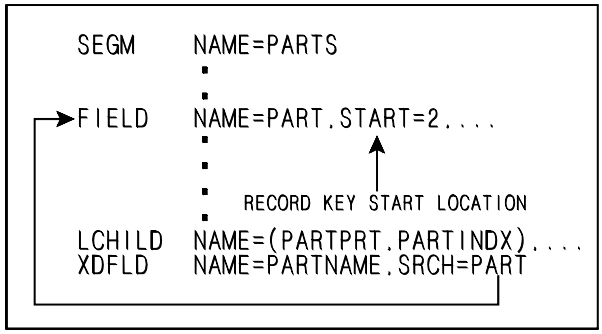



Figure 3-44 Virtual Key Start Location

7. The length of the record key is the combined length of all fields specified by the SRCH keyword of the XDFLD statement. Individual field lengths can be determined by finding each field's BYTES keyword value among the FIELD statements for the index source segment type, which is the segment named by the SEGMENT keyword of the XDFLD statement. The combined length of these fields is used as the length of the virtual key field, which must have a blank field location. See Figure 3-45 for an example.

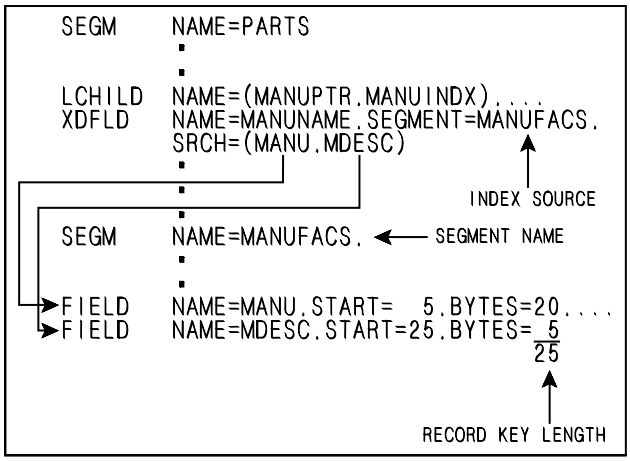



Figure 3-45 Virtual Key Length

8. The field type for the record key is always a character string. This is true regardless of the characteristics of the individual fields that make up the secondary index.

When the PCB specifies the INDEXES keyword of the SENSEG keyword for a root segment, the VISION:Builder file definition can be prepared exactly as described in the preceding guideline. The only difference between using the INDEXES keyword at the root level and using the PROCSEQ keyword is the root segments will be retrieved in primary processing sequence instead of secondary sequence. Random processing for both will be by secondary index specified on the XDFLD statement.

However, when the PCB specifies the INDEXES keyword for lower level segments, some processing problems can arise. Although it is possible to define a secondary index as a lower level segment key, it is not recommended. Instead, the file definition should define normal segment keys for all lower level segments for which the INDEXES keyword is specified.

# Defining a Relational File

## Understanding Logical Records

A logical record is a hierarchical record that is defined in a VISION:Builder file definition. Historically, there has been a one-to-one correspondence between a logical record and a physical record and so there was little need to make a distinction. But with the advent of relational databases, the logical record takes on a new meaning, as it can be composed of data from many physical sources (tables). The logical record, in effect, joins data from tables in such a way that you can construct a record that does not physically exist anywhere in a database.

The VISION:Builder file definition describes the logical record structure and provides the information that enables VISION:Builder to initialize the logical data from the fields and tables that comprise the physical data. Therefore, the relational data is mapped into a *logical record*.

VISION:Builder operates by copying physical data from disk into memory, making all extractions and changes of data, and copying the changed data back to the disk. The record in memory against which all the activity takes place is really a *logical record*.

Because VISION:Builder operates against logical records, the VISION:Builder specifications and processing flow are independent of the particular database used to construct the logical record. Therefore, VISION:Builder is used to develop applications using relational databases in exactly the same way that it is used with VSAM or IMS.

Support of a database access method (relational in particular) revolves around mapping the physical data from disk into the required logical representation to be used by the application. Once you have the logical record, all the rules that you have learned about looping, request processing, file coordination, MOSAIC processing, and report generation apply.

## Advantages of Logical Records

The logical record facility is a capability unique to VISION:Builder (and other Computer Associates products). This capability enables you to develop sophisticated applications both rapidly and accurately.

The logical record capability when applied to relational tables provides you with the following capabilities and advantages:

### Improved Problem Solving

In using relational databases, you are not required to think of your data as a set of disjointed tables but can regard the data as a set of data integrated and structured in a manner that is suitable for solving the problem at hand. In short, you need not worry about how the data is stored. You are at liberty to define the data to be the way that you *want* it was stored.

The same set of relational tables can be viewed as different logical structures. Problem solving is greatly facilitated if the data is organized in a convenient manner. The logical record capability of VISION:Builder enables you to always have the data organized in the most convenient way.

### Conceptual View

The logical record enables you to think of the problem within an appropriate framework that makes it understandable. Relational tables are a good way to organize sets of related data, but they do not lend themselves to increasing the understanding of what is going on.

This concept is illustrated by the example described in Example 1 Using a Logical Record on page 3-55. In that example, there are two tables, the Department table and the Employee table. The application is designed to organize employees by department. Conceptually, it is much easier to look at this example as a department root, with all of the employees subordinate to it, than to understand the concept of separate tables joined on matching values. Example 2 Creating an Organization Chart on page 3-59 provides an interesting example of the ability to understand the problem by formulating it appropriately.

### Focus on the Business Problem

In solving a problem and developing an application, you, as the application developer, are faced with the dual problems of implementing the business requirements with all of the sequences and algorithms involved, as well as working with the data organizations to ensure that the correct data is available when required.

The logical record capability enables you to define your data requirements completely independent of the business application proper. You can now forget about the data and get on with solving the real problem. VISION:Builder guarantees that the right data will be available when it is needed.

### Automatic Navigation

In defining a logical record, you are defining all of the relationships between the data. In developing the application proper, it is only necessary to reference fields by name and VISION:Builder will automatically navigate through the tables to ensure that all of the correct combinations of data are presented to your application for processing.

### Views and Security

You create logical relationships to relate data or to filter data so that the application sees only a relevant subset. For example, in Example 1 Using a Logical Record on page 3-55, you could have screened the data by selecting only the root segment (DEPTNO) equal to B01 (Sales). It would be as if the other departments (Finance and Engineering) did not exist. Only the logical record for the Sales department would be accessed.

This capability limits the scope of the data to the data that is relevant to the situation, imposing security upon the data by limiting users to data for which they have a need to know.

### Referential Integrity

The referential integrity that is inherent in the VISION:Builder logical record does not interfere with the referential integrity capability provided by DB2 or other relational database managers.

For example, you want to remove the Sales department record from the example in Example 1 Using a Logical Record on page 3-55. The action of deleting the Sales root segment automatically deletes the root segment from the logical record and

the row from the Department table. The subordinate segments are deleted only from the logical record. The rows in the Employee table are deleted only if the relational database manager determines, through its referential integrity rules, that the deletes are required.

VISION:Builder is responsible for maintaining the logical record. The relational database manager is responsible for maintaining the tables. Referential integrity is described in greater detail in Chapter 15, *Advanced Transaction Processing*.

### Processing Efficiency

The logical record capability (with the ability to subset the fields retrieved from the tables combined with the ability to filter the data) enables VISION:Builder to perform the minimum number of database accesses to retrieve the data for the application. Expensive DB2 joins are avoided.

For example, the Employee table lists thousands of employees but only five are listed for department B01. You could limit processing on the root segment to department B01 only, resulting in the database manager passing only five records to VISION:Builder. This is much more efficient than having VISION:Builder inspect and discard thousands of records.

### Data Accuracy (or Outer Joins)

The logical record facility is actually the equivalent of an outer join in relational terms. An outer join occurs when data exists in one table but there is no corresponding data in the related table. A normal join will ignore this situation and the application will never see it. This could provide a very misleading set of results and lead to incorrect corporate decisions being made.

An outer join is normally quite difficult for COBOL, PL/I, or other programs to achieve and it is not native to DB2. The outer join is the normal mode of operation for VISION:Builder, ensuring that all relevant data is made available to the application.

For example, you want to add a Legal department to the Department table. Currently, the Legal department is not staffed. Because there are no employees, there are no subordinate segments.

VISION:Builder is able to update this logical record (and subsequently the physical tables) as employees are hired for this department. No extra work is required for you in developing your VISION:Builder application. It is all taken care of automatically. Any application developed in a conventional third-generation language utilizing a join (the normal way of solving this kind of problem) is unable to handle this boundary situation.

### Preservation of Concepts and Applications

The logical record concept preserves all of your VISION:Builder training and expertise. Looping, transaction processing,❹ file coordination, procedure processing, and so on, all work in exactly the same manner as they do for IMS and

VSAM. This means that applications developed for IMS work with relational tables without any source changes to the report specifications or application logic. change the file definitions to map the tables into logical records.

The logical record provides you with complete independence from the physical structure of the database. You solve a problem by integrating and structuring the data in a manner most convenient for your application.

## Defining the Logical Record

It is evident that the key to the support of relational databases is that of defining logical records and mapping the data from the tables into the records. This is done through the file definition.

Logical records are defined to VISION:Builder in exactly the same way as for VSAM and IMS. There are some new extensions to the file definition to establish the physical mapping (that is, the physical data that is to be moved into the logical record structure).

Within a relational VISION:Builder file definition, every segment represents a table. This is illustrated in Figure 3-46.

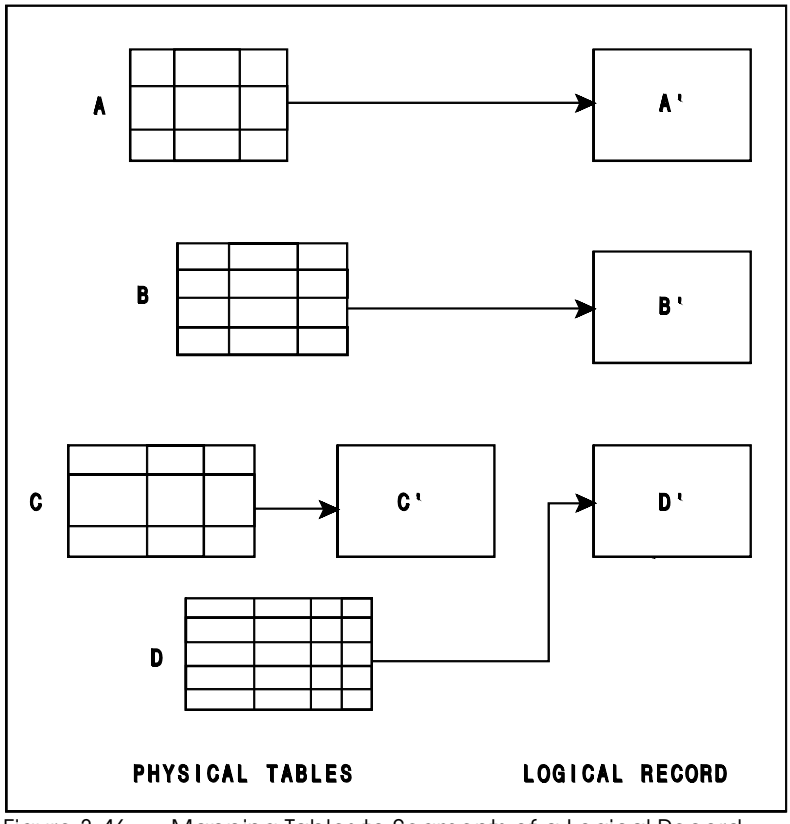



Figure 3-46 Mapping Tables to Segments of a Logical Record

Figure 3-46 shows that logical segment A' represents the table A, B' represents B, and so on. The only specification required to complete this mapping is to identify to VISION:Builder which table the segment is to represent.

Every segment type represents a table; every segment occurrence in the logical record represents a row (or partial row) of the corresponding table.

Because the segment is logical, it does not have to contain all of the columns of the relational table that it represents. This differs from VSAM and IMS, where the logical and the physical must be the same. This is illustrated in Figure 3-47.

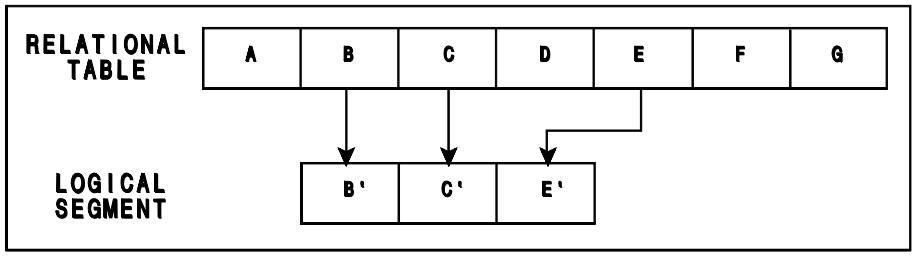



Figure 3-47 Subsetting the Table Columns

Figure 3-47 shows the mapping (the rules for movement) of physical fields B, C, and E to logical fields B', C', and E'. When defining the logical field, you merely identify which physical field is to supply the data to fill it.

This automatic mapping capability provides more than mapping a single physical data field into a logical data field. It is possible to map scalar functions such as:

```
1.05   * SALARY
BUDGET + VARIANCE
```

Any legitimate scalar function supported by DB2 can be used for the purpose of mapping into the logical record.

In addition to scalar functions, it is also possible to use the DB2 built-in column functions. For example:

```
MAX   (SALARY)
MIN   (SALARY)
AVG   (SALARY)
SUM   (BUDGET)
```

Here again, any legitimate function can be used.

A logical record can be comprised of fields that are derived from the physical data. A logical data field can actually be a copy of the physical field, the result of an arithmetic expression or a compound field obtained by summarizing over several rows of the table.

With relational databases, the physical is completely divorced from the logical. This allows you to have unmapped fields in the logical segment. Unmapped fields are fields in the segment that are not filled with data from a relational table. They can be used as temporary working fields associated with the segment.

Unmapped fields are illustrated in Figure 3-48.

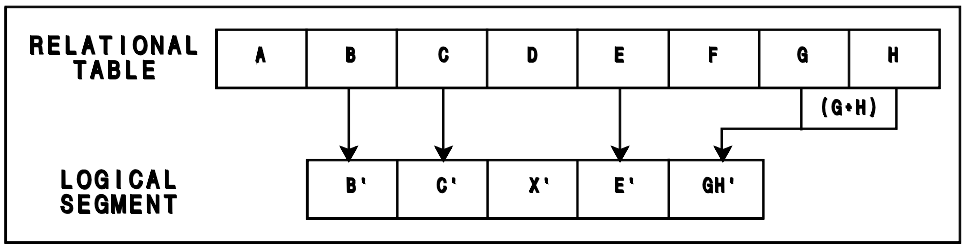



Figure 3-48 Defining Unmapped Fields

In Figure 3-48, the logical field X' exists for the use of the application, but does not exist in the database. The field GH' illustrates the mapping of a scalar function (that is, G+H).

A logical segment consists of a set of mapped fields that are filled with data from columns of a single relational table and a set of unmapped fields that are initialized to blank or zero, as appropriate. Logical fields can overdefine other logical fields. This generally only makes sense when an unmapped field overdefines some mapped fields. You might have DATE (unmapped, six characters) overdefining the mapped fields of MONTH, DAY, and YEAR (all mapped as two characters).

In IMS and VSAM, the correct parentage between segments is established by the position of the data in the record. With relational tables, the separate tables have no positional relationship. You must provide explicit rules or logical relationships that establish which child segment belongs to which parent segment. These rules can also provide criteria for filtering the segments to be selected. Each row satisfying the relationship criteria is selected from the table and is mapped into a segment.

The process of constructing a logical record is done one segment at a time. First a root is obtained. Subordinate segments are added if they satisfy the rules or logical relationships.

## The Relational File Definition

This chapter has described the concept of the logical record, the process of mapping a (physical) row into a (logical) segment and establishing a hierarchical structure through logical relationships or rules. All of these are defined to VISION:Builder in a relational file definition. You define a relational file definition just like you would for other file types with the following exceptions:

| In the File Definition Window: | In the Segment Definition Window: |
|---|---|
| ■ Specify the Record Format as relational.<br>■ Establish the logical relationship of each dependent segment to its parent in the Logical Relation popup window. | ■ Associate a relational table name with each segment.<br>■ Associate a relational table column name with each field that is to be mapped.<br>■ All key fields must be mapped, unless the segment is a multi-column keyed segment. In this case, each component of the key must be mapped. See Example 11 Defining a Multi-Column Keyed Segment on page 3-86 for details on multi-column keyed segments. |

As an example, review the Department table and the Employee table shown in Figure 3-49.

```
DEPARTMENT_TABLE

    DIVISION       DEPTNO    NAME          MANAGER_NO

    WIDGETS        A01       FINANCE         123
    WIDGETS        B01       SALES           456
    WIDGETS        C02       ENGINEERING     789


 EMPLOYEE_TABLE

    EMPNO     NAME    CLASS    SALARY    DEPT     LOC

    231       JONES     23     25000     B01      L65
    234       SMITH     34     45000     B01      L80
    123       WILSON    45     75000     A01      L65
    477       BROWN     29     35000     B01      L80
    456       WHITE     34     45000     B01      L75
    333       GREEN     44     25000     C02      L80
    789       BLACK     44     67000     C02      L80
    567       BLUE      23     25000     A01      L65
    444       ADAMS     23     25000     B01      L65
```

Figure 3-49 Department-Employee Example, Physical Data

These are two separate tables, a collection of departments and a collection of employees. For this example, however, view them as a logical record with the root segment as the department and the subordinate segment as the employees within the department. The File Definition window in Figure 3-50 shows the hierarchy for the logical record.

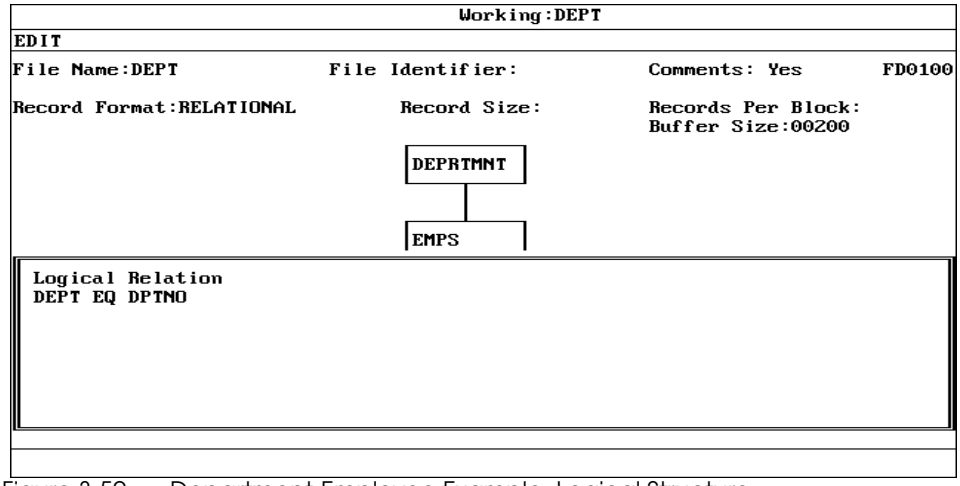



Figure 3-50 Department-Employee Example, Logical Structure

The logical relationship between the EMPS child segment and the DEPRTMNT parent segment is shown in the popup window. This relationship transforms rows of the Employee table into employee segments only if the DEPT column of the EMPLOYEE_TABLE matches the field DPTNO of the root segment.

In general, a logical relationship is entered as a logical expression. The left hand operand is any relational table column name and need not be defined as a field in the logical record. It is followed by a relational operator (EQ, NE, LT, and so on). The right hand operand is either a constant or a VISION:Builder field name defined in this file definition in any ancestor segment. The logical relationship can test for multiple conditions by combining logical expressions with AND or OR connectors.

**Note:** The tables to the left of Figure 3-51 through Figure 3-101 describe the relationship between the Builder Field Name (shown in the figures) and its corresponding DB2 Column Name (not shown), by Segment Name. To find the DB2 Column Name, enter Y in the More Attr column (shown in the figures) and press F8.

The Segment Definition windows in Figure 3-51 show the table name and column names for each segment.

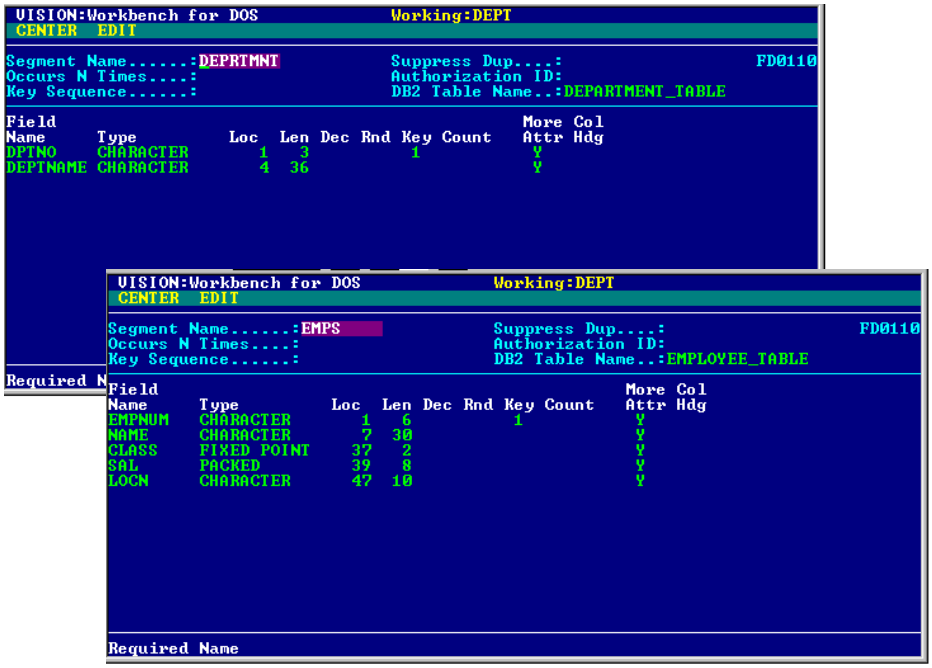



Figure 3-51 Department-Employee Example, Mapping Specification

**Segment Name:** DEPRTMNT

| Builder Field Name | DB2 Column Name |
|---|---|
| DPTNO | DEPTNO |
| DEPTNAME | NAME |

**Segment Name:** EMPS

| Builder Field Name | DB2 Column Name |
|---|---|
| EMPNUM | EMPNO |
| NAME | NAME |
| CLASS | CLASS |
| SAL | SALARY |
| LOCN | LOC |

The DB2 Name entry contains the relational table name for the segment. The column name associated with each mapped field is shown in the popup windows to the right of the Segment Definition window. The Y under DB2 Name indicates that the field is a mapped field. Although this example does not contain any unmapped fields, they would be specified by merely omitting the DB2 Name for the unmapped fields.

Note that VISION:Builder allows a different scope of field types from DB2. You do not need to worry about this; in the transference of data from the table to the segment (and vice versa), conversions for compatibility take place automatically.

The difference can be relevant to the extent that it affects performance. In short, where possible, it is best to have the VISION:Builder field specification be the same as the relational column specification.

**Note:** A value of blanks in a Date/Time field is not valid in DB2.

**Note:** The VISION:Builder length of a GRAPHIC and LONG GRAPHIC string is (2*n), where n is the DB2 length of the string.

### Guidelines for Defining VISION:Builder Field Types

| DB2 Type, Length | | VISION:Builder Type, Length | |
|---|---|---|---|
| CHAR | 1-254 | C | 1-255 |
| VARCHAR | 1-254 | V | 1-254 |
| To update DB2 tables in-place while using VARCHAR fields, use non-MOSAIC processing, which only allows update-in-place processing on a single table. Therefore, if you need to hierarchically join two or more tables, then you cannot use VARCHAR fields. | | | |
| LONG VARCHAR | 255 and larger | V | 255-99H |
| DECIMAL | | P | 1-15 |
| SMALLINT | | F | 1-2 |
| INTEGER | | F | 3-4 |
| FLOAT | | E | 4 |
| DATE | | C | 10 |
| TIME | | C | 8 |
| TIMESTAMP | | C | 16-26 |
| GRAPHIC | 1-127 | C | 2-254 |
| LONG GRAPHIC | 128 and larger | V | 256-99H |

The process of constructing a logical record is done one segment at a time. First, a root segment is obtained. Subordinate segments are added if they satisfy the rule or logical relationship. In this case, rows of the Employee table must be transformed into the employee segments only if the column DEPT of the EMPLOYEE_TABLE matches the field DPTNO of the root segment. In other words, having picked a root segment (and, therefore, having obtained a value for DPTNO in the logical record), the second table will only pass information if there is a match.

In this relational file definition, the VISION:Builder application is written against the logical record, not the separate relational tables. Using the sample data defined in Figure 3-49 on page 3-49, the VISION:Builder application will process the records shown in Figure 3-52.

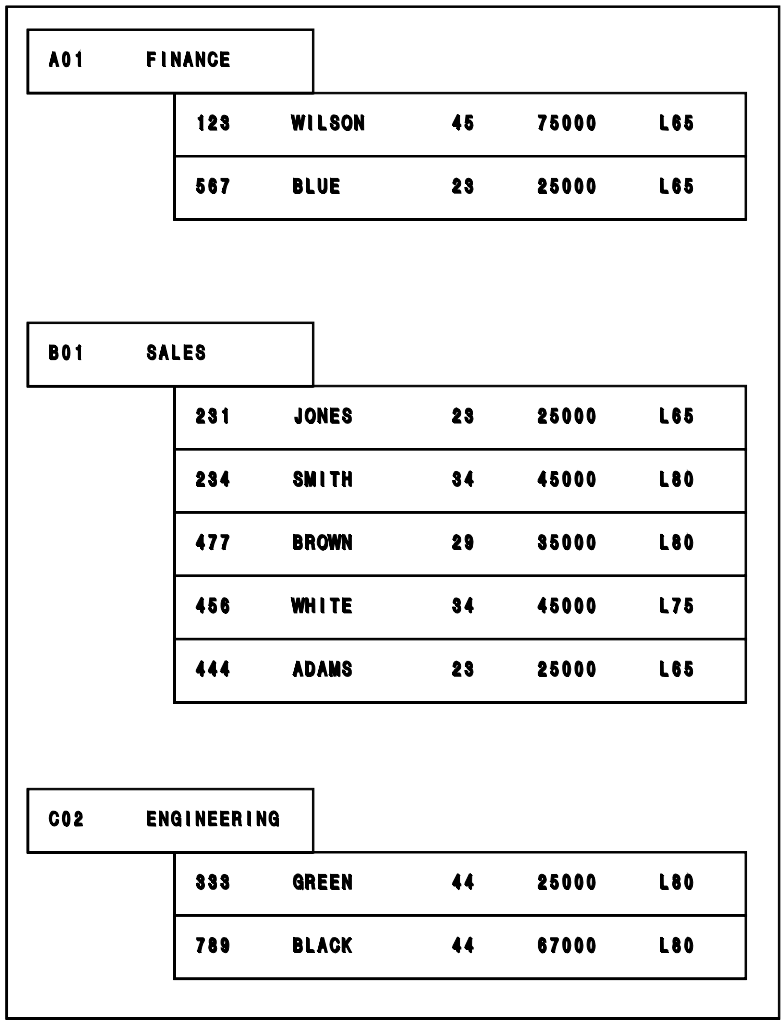



Figure 3-52 Department-Employee Example, Logical Record

The logical record can contain summarized data that is derived by use of column functions. Consider a record that contains the employee information for the highest paid employee in each department. The logical structure is shown in Figure 3-53.

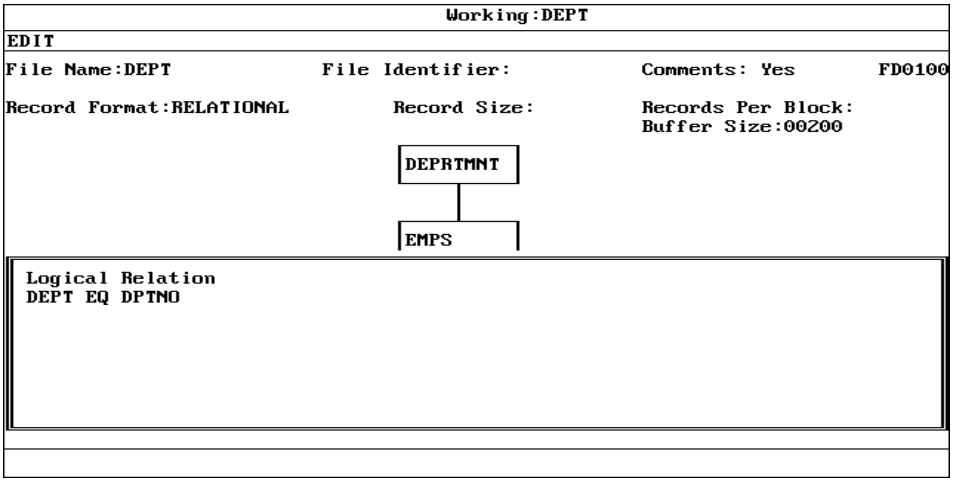



Figure 3-53 Highest Paid Employee Example, Logical Structure

The mapping specification is shown in Figure 3-54.

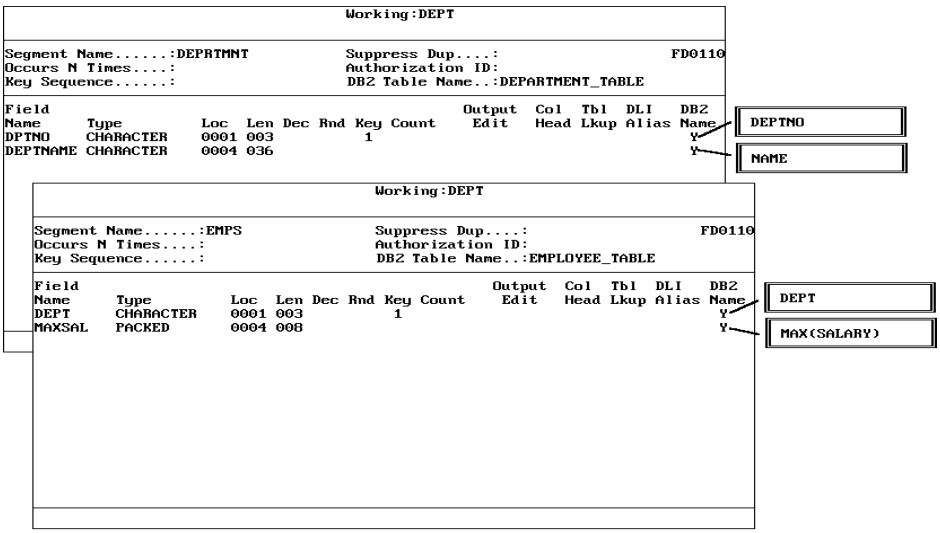



Figure 3-54 Highest Paid Employee Example, Mapping Specification

The process of constructing a logical record is done one segment at a time, but a subordinate segment is added for each value of the column function returned by DB2.

Using the sample data defined in Figure 3-49, the VISION:Builder application will process the records shown in Figure 3-55.

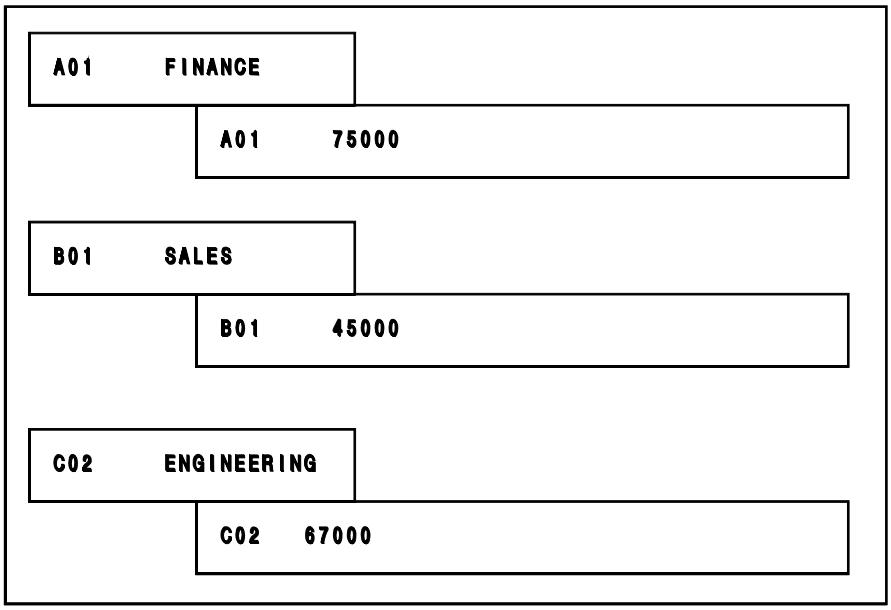



Figure 3-55 Highest Paid Employee Example

While these records do not physically exist, VISION:Builder makes them exist for the purposes of the application.

The logical record approach allows for the integration of up to 99 relational tables or relational views (one per segment) into a single logical record. This makes for a maximum of 990 tables or views in a VISION:Builder application (the master file and nine coordinated files), which is far more than the ten available if each VISION:Builder file is treated as a single relational table or view. In each logical segment, up to nine key fields can be specified.

The following sections in this chapter contain specific examples of relational tables and the logical records that can be defined for them.

## Example 1 Using a Logical Record

The logical record concept enables you to define the relationships between tables prior to generating the required report.

Suppose you have three relational tables: DIVISION_TABLE, DEPARTMENT_TABLE, and EMPLOYEE_TABLE. You want to report all the information for each employee and organize it by division and department. To implement the application, imagine you have a file in exactly the structure that you need. Assume a file with a logical structure as shown in Figure 3-56.

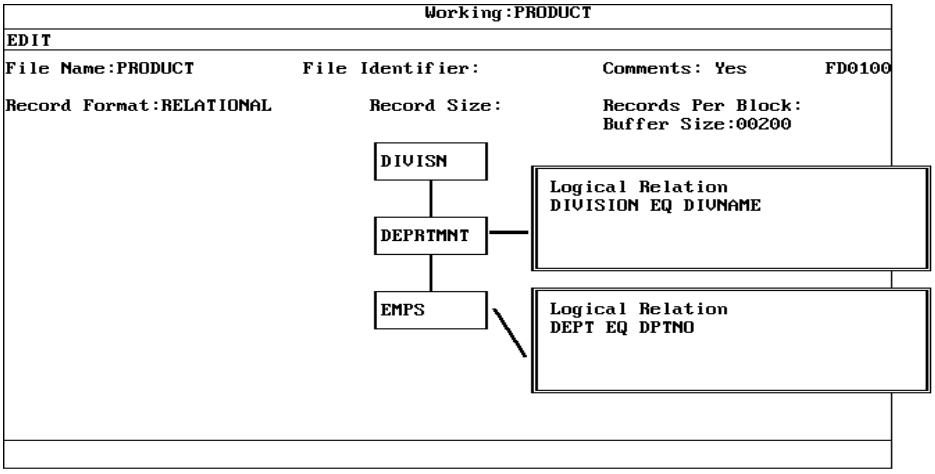



Figure 3-56 Example 1: Logical Structure

**Segment Name:** DIVISN

| Builder Field Name | DB2 Column Name |
|---|---|
| DIVNAME | DIV |
| REGION | REGION |

**Segment Name:** DEPRTMNT

| Builder Field Name | DB2 Column Name |
|---|---|
| DPTNO | DEPTNO |
| DEPTNAME | NAME |

**Segment Name:** DEPRTMNT

| Builder Field Name | DB2 Column Name |
|---|---|
| EMPNUM | EMPNO |
| NAME | NAME |
| CLASS | CLASS |
| SAL | SALARY |
| LOCN | LOC |

The mapping specification is shown in Figure 3-57.

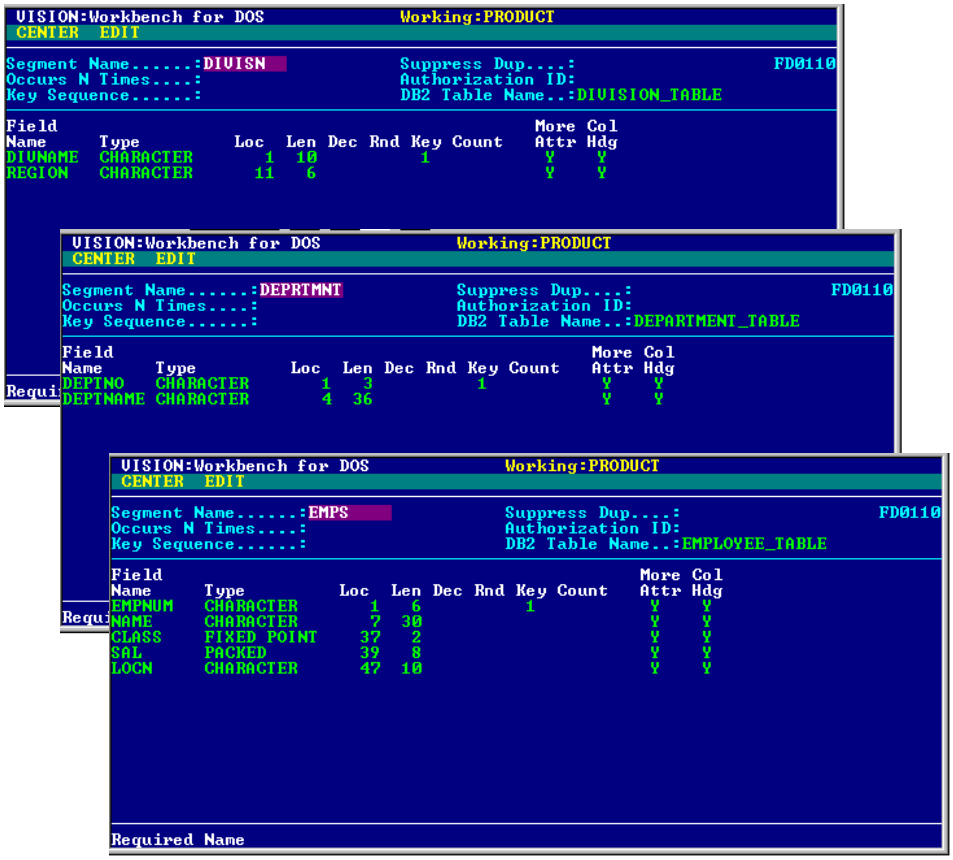



Figure 3-57 Example 1: Mapping Specification

Each segment represents the table whose name appears in the DB2 Table Name entry. Because each table has a different owner, the authorization ID of the owner is entered in the Authorization ID entry. You define only the fields that are of interest to the application. For example, DIVISION from the DEPARTMENT_TABLE and DEPT from the EMPLOYEE_TABLE are not included because these fields are not actually reported by the application.

The logical relationship provides the rule by which a row is retrieved from a table to become a subordinate segment of a previously retrieved parent segment. You can think of the process as if VISION:Builder first retrieves the root segment and assembles the logical record level by level. Decisions concerning which rows of a table are to be added to the logical record are made based upon the segments already constructed (that is, rows already retrieved).

In Figure 3-56 on page 3-56, the logical relationship for the DEPRTMNT segment provides the rule: Include as a subordinate DEPRTMNT segment any row from the DEPARTMENT_TABLE where the column DIVISION has the same value as the logical field DIVNAME in the parent segment.

The logical relation for the EMPS segment provides the rule: Include as a subordinate EMPS segment any row from the EMPLOYEE_TABLE where the column DEPT has the same value as the logical field DPTNO in the parent segment.

Using sample data, a typical logical record is shown in Figure 3-58.

```
WIDGETS  EAST
     A01 FINANCE
          123     WILSON    45     75000     L65
          567     BLUE      23     25000     L65
     E05 LEGAL
     B01 SALES
          231     JONES     23     25000     L65
          234     SMITH     34     45000     L80
          477     BROWN     29     35000     L80
          456     WHITE     34     45000     L75
          444     ADAMS     23     25000     L65
     C02 ENGINEERING
          333     GREEN     44     25000     L80
          789     BLACK     44     67000     L80
```

Figure 3-58 Example 1: Logical Record

In other words, the VISION:Builder application treats this record as if it actually exists. The Report Detail window from the application is shown in Figure 3-59.

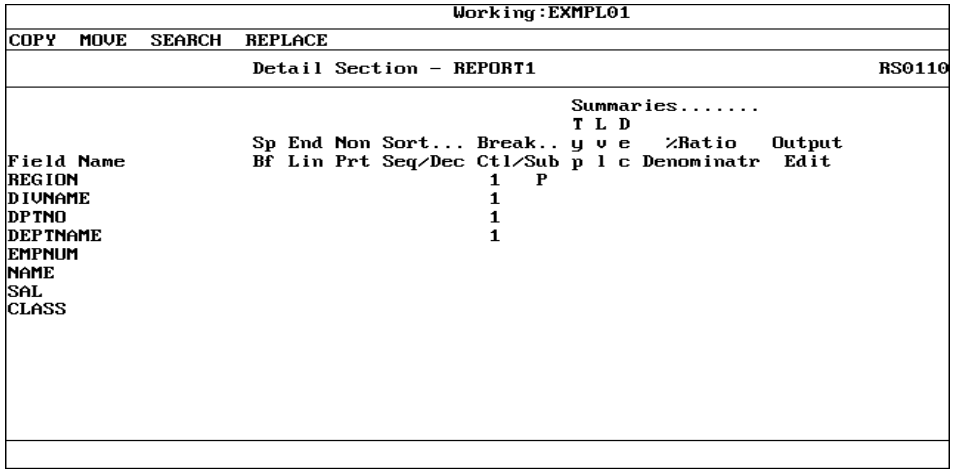



Figure 3-59 Example 1: Report Detail Window

The report produced by the application is shown in Figure 3-60.

```
                        REPORT OF EMPLOYEES BY DEPARTMENT
EAST
------------------------------------------------------------------------------
   DIVISION     DEPT      DEPT         EMPLOY     NAME      SALARY     CLASS
                NUMBER    NAME         NUMBER
------------------------------------------------------------------------------
   WIDGETS      A01       FINANCE      123        WILSON    $75,000     45
                                       567        BLUE       25,000     23

   WIDGETS      E05       LEGAL         -          -          -          -
```

Figure 3-60 (Page 1 of 2)Example 1: Report Output

```
WIDGETS      B01      SALES          231        JONES     $25,000   23
                                     234        SMITH      45,000   34
                                     444        ADAMS      25,000   23
                                     456        WHITE      45,000   34
                                     477        BROWN      35,000   29

WIDGETS      C02      ENGINEERING    333        GREEN     $25,000   44
                                     789        BLACK      67,000   44
```

Figure 3-60 (Page 2 of 2)Example 1: Report Output

Using logical records, the application becomes clearer and simpler to code. In this example, you do not accept the data as it is; you define it to address the application.

## Example 2 Creating an Organization Chart

In this example, assume only one table, a personnel table containing a row for each employee. The table also contains each employee's manager's name. Assume that the president has no manager; therefore, the MANAGER_NAME column is NULL. The company hierarchy is:

- President
- Vice President
- Manager
- Technical Staff

To develop an application that produces an organization chart, create a logical structure as shown in Figure 3-61.

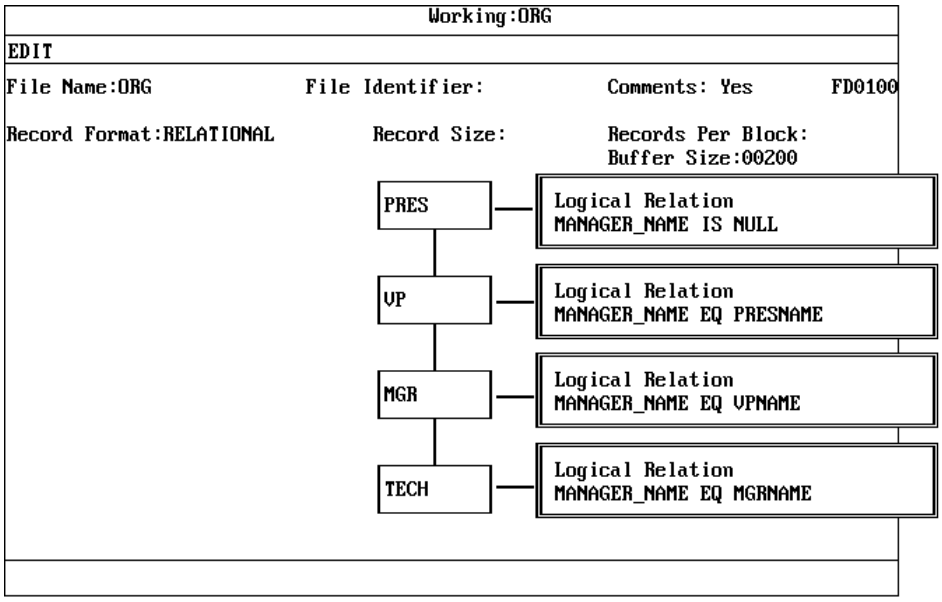



Figure 3-61 Example 2: Logical Structure

**Segment Name:** PRES

| Builder Field Name | DB2 Column Name |
|---|---|
| PRESNAME | EMPLOYEE_NAME |
| PRESSAL | EMPLOYEE_SALARY |

**Segment Name:** VP

| Builder Field Name | DB2 Column Name |
|---|---|
| VPNAME | EMPLOYEE_NAME |
| VPSAL | EMPLOYEE_SALARY |

**Segment Name:** MGR

| Builder Field Name | DB2 Column Name |
|---|---|
| MGRNAME | EMPLOYEE_NAME |
| MGRSAL | EMPLOYEE_SALARY |

**Segment Name:** TECH

| Builder Field Name | DB2 Column Name |
|---|---|
| TECHNAME | EMPLOYEE_NAME |
| TECHSAL | EMPLOYEE_SALARY |

The mapping specification is shown in Figure 3-62.

Figure 3-62 Example 2: Mapping Specification

Here, the same table is used with the same fields to form a four level hierarchy. It is the logical relations that make sure that the right rows are inserted into the right levels. The logical relation for the root segment establishes that the president is the individual with no manager (the test is that the MANAGER_NAME column is NULL).

For example, take the sample data shown in Figure 3-63.

| EMPLOYEE_NAME | MANAGER_NAME | SALARY |
|---|---|---|
| WATERS | SIMPSON | 65000 |
| HILL | WATERS | 50000 |
| ANDREWS | SIMPSON | 60000 |
| WISE | THOMAS | 35000 |
| MILLS | THOMAS | 26000 |
| MARTIN | WATERS | 45000 |
| SIMPSON | (NULL) | 85000 |
| SMITH | MARTIN | 33000 |
| JACKSON | MARTIN | 25000 |
| KRULL | HILL | 30000 |
| THOMAS | ANDREWS | 35000 |
| ARNOLD | MARTIN | 31000 |

Figure 3-63 Example 2: Sample Data

Figure 3-64 shows the sample data as a logical record.

```
SIMPSON         85000
     WATERS         65000
          MARTIN        45000
              SMITH          33000
              ARNOLD         31000
              JACKSON        25000
          HILL          50000
              KRULL          30000
     ANDREWS        60000
          THOMAS        35000
              MILLS          26000
              WISE           35000
```

Figure 3-64 Example 2: Logical Record

The four logical relations, as shown in Figure 3-62, are all that are required to construct the complete organization hierarchy. The application report detail is shown in Figure 3-65.

```
                                        Working:EXMPL02
COPY  MOVE  SEARCH  REPLACE
                    Detail Section - REPORT1                                RS0110

                                                Summaries.......
                                                T L D
                    Sp End Non Sort... Break.. y v e   %Ratio   Output
Field Name          Bf Lin Prt Seq/Dec Ctl/Sub p l c Denominatr  Edit
PRESNAME                                 1
VPNAME                                   2
MGRNAME                                  3
TECHNAME
```

Figure 3-65 Example 2: Report Detail Window

Figure 3-66 shows the resulting organization chart.

```
                    ORGANIZATION CHART
-----------------------------------------------------------
      PRESIDENT     V.P.    MANAGER      EMPLOYEE
-----------------------------------------------------------
     SIMPSON       WATERS   MARTIN       SMITH
                                         ARNOLD
                                         JACKSON

                            HILL         KRULL

                   ANDREWS  THOMAS       MILLS
                                         WISE
```

Figure 3-66 Example 2: Output

In this example, without the logical record capability, you could have used a procedural approach, but that could have taken you several days or weeks and several computer runs to solve with a third-generation language.

The logical record approach is a non-procedural approach to the solution that can be developed in less than 15 minutes and will run the first time. Notice that once you have defined the data the way that you want it, the job is over. The entire application consists mainly of report detail specifications.

### Example 3 Mapping Tables into a Logical Record

When mapping a record from a group of tables, you can use the orderly nature of relational tables and still map the information into a logical record that better suits your application. A group of normalized tables can be well-focused and easy to maintain, but they can lead to several tables that point to other tables to connect all the information.

This example illustrates the mapping of a customer order structure that logically connects all the information from several tables. Notice that the primary order information is contained in two separate tables: CUST_ORDER and ORDER_TABLE_2. All the tables used in the mapping of the logical record are shown in Figure 3-67.

```
CUSTOMER_TABLE
CUSTOMER_NO, CUSTOMER_NAME, CUSTOMER_PHONE

CUST_ORDER_TABLE
CUSTOMER_ NUMBER, ORDER_NUMBER, PO_NUMBER

ORDER_TABLE_2
ORDER_NUM, ORDER_DATE, AGENT

ORDER_ITEM_TABLE
ORDER_NO, ITEM_NUMBER, QUANTITY

ITEM_TABLE
ITEM_NO, ITEM_NAME, ITEM_PRICE
```

Figure 3-67 Example 3: Table Information

The logical structure is shown in Figure 3-68.

**Note:** In Figure 3-68, the root segment, CUSTOMER, is present but not depicted.

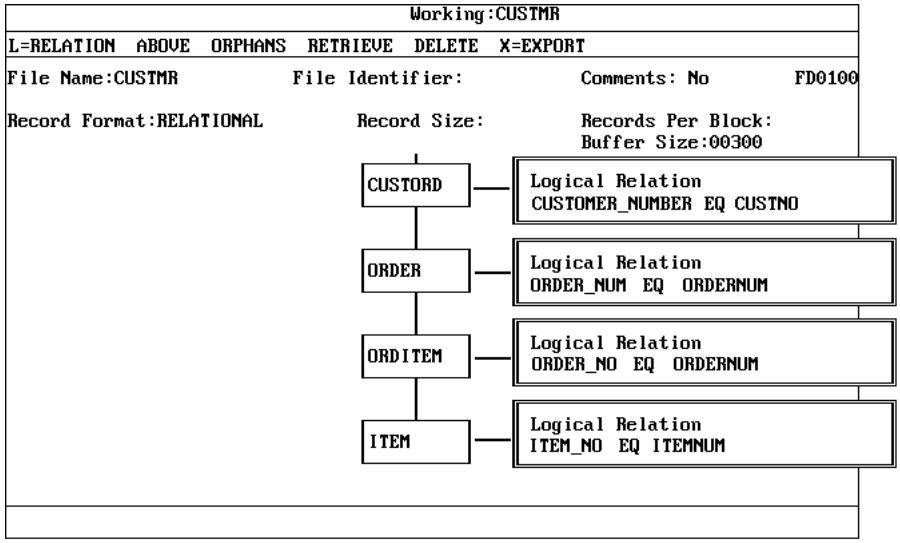



Figure 3-68 Example 3: Logical Structure

**Segment Name:** CUSTOMER

| Builder Field Name | DB2 Column Name |
|---|---|
| CUSTNO | CUSTOMER_NO |
| CUSTNAME | CUSTOMER_NAME |
| CUSPHONE | CUSTOMER_PHONE |

**Segment Name:** CUSTORD

| Builder Field Name | DB2 Column Name |
|---|---|
| ORDERNUM | ORDER_NUMBER |
| PONUM | PO_NUMBER |

**Segment Name:** ORDER

| Builder Field Name | DB2 Column Name |
|---|---|
| ORDATE | ORDER_DATE |
| AGNT | AGENT |

**Segment Name:** ORDITEM

| Builder Field Name | DB2 Column Name |
|---|---|
| ITEMNUM | ITEM_NUMBER |
| ITEMQTY | QUANTITY |

**Segment Name:** ITEM

| Builder Field Name | DB2 Column Name |
|---|---|
| ITEMNAME | ITEM_NAME |
| PRICE | ITEM_PRICE |

The mapping specification is shown in Figure 3-69.

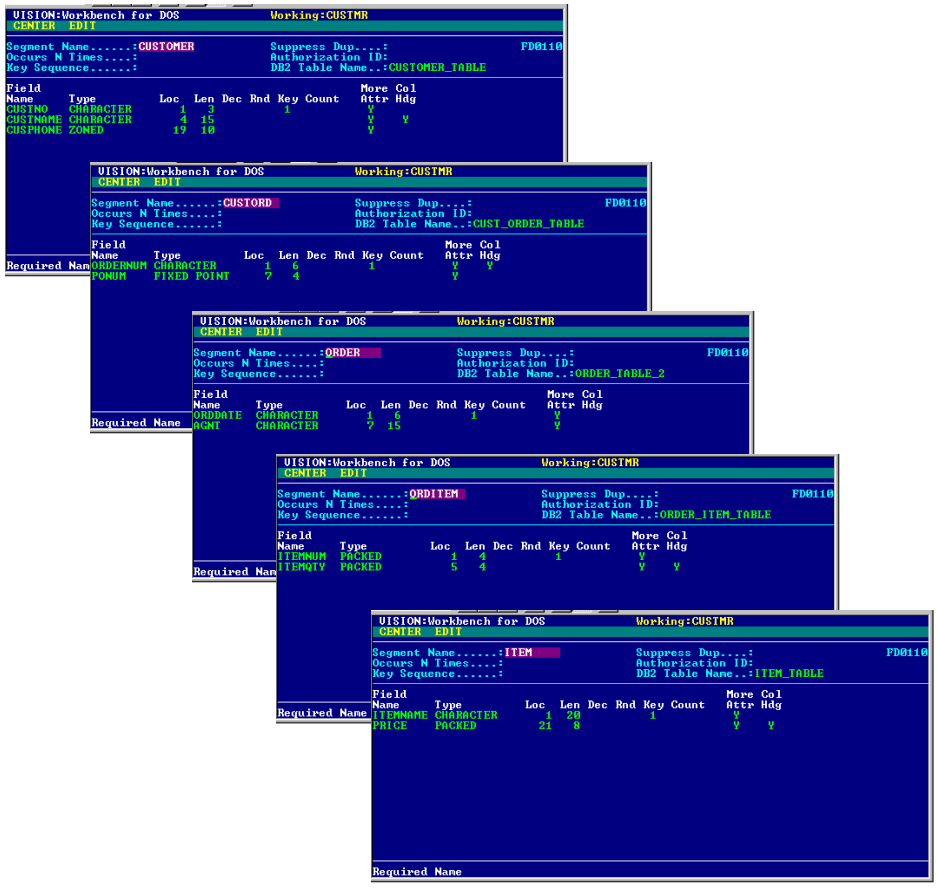



Figure 3-69 Example 3: Mapping Specification

It is evident, in this example, that a single occurrence of the CUSTORD segment has a single ORDER child segment (true also of the ORDITEM and ITEM segments). See Figure 3-68 on page 3-64. An occurrence of a segment with a single child is equivalent to the concatenation of the two segments. More tables lead to a greater depth of the logical record.

Figure 3-68 illustrates how the tables are joined together to form a logical record. The logical relationship for the ORDITEM segment shows that this relationship does not have to depend on its immediate parent (ORDERNUM is in the CUSTORD segment). Logical relationships are generally formulated in terms of the immediate parent, but any ancestor segment will do. In fact, logical relationships do not even have to be formulated using variable data.

## Example 4 Viewing the Customer-Order Database by Item

The previous example viewed the customer-order database from the conceptual viewpoint of the customer as the root segment. This example focuses more on items than customers. Suppose that you would like to keep track of which customers had purchased a specific item. Assume you are using the tables as defined in Figure 3-70.

```
CUSTOMER_TABLE
CUSTOMER_NO, CUSTOMER_NAME, CUSTOMER_PHONE

ORDER_TABLE
CUSTOMER_NUM, ORDER_NUMBER, ORDER_DATE, PO_NUMBER, AGENT

ORDERED_ITEMS_2
ORDER_NO, ITEM_NAME, ITEM_PRICE, QUANTITY
```

Figure 3-70 Example 4: Table Information

As mentioned, the logical data view capability of VISION:Builder gives you the ability to define the data in the way that you would like it to be. For this application, the data has the structure as shown in Figure 3-71.

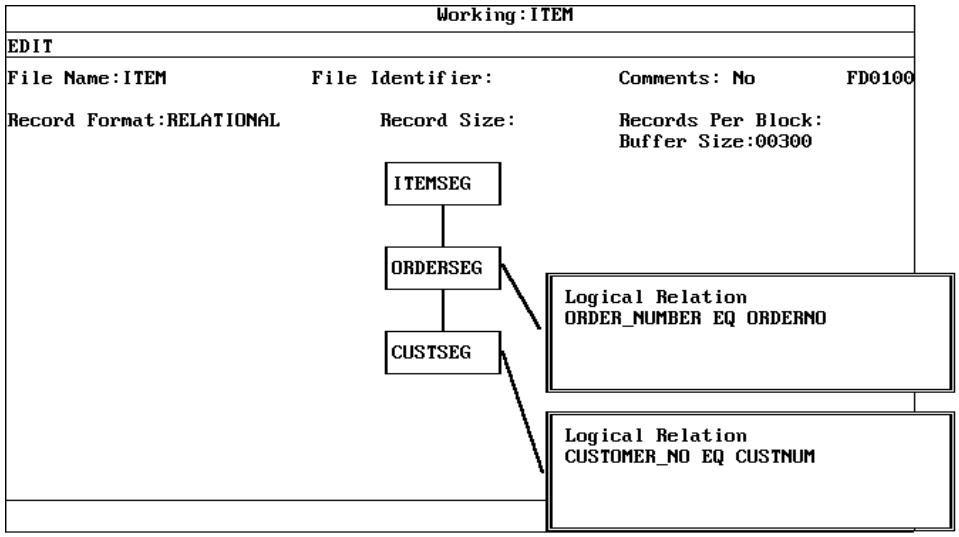



Figure 3-71 Example 4: Logical Structure

This gives you a series of records with item as the root segment. This approach generates a series of records with duplicate keys, one for each separate order. This logical file can be processed if the Sequence Check (Chk) entry in the Files Used window is set to U (for unsequenced) or E (for equal keys).

**Segment Name:** ITEMSEG

| Builder Field Name | DB2 Column Name |
|---|---|
| ITEMNAME | ITEM_NAME |
| ORDERNO | ORDER_NO |

**Segment Name:** ORDERSEG

| Builder Field Name | DB2 Column Name |
|---|---|
| CUSTNUM | CUSTOMER_NUM |

**Segment Name:** CUSTSEG

| Builder Field Name | DB2 Column Name |
|---|---|
| CUSTNAME | CUSTOMER_NAME |

The mapping specification is shown in Figure 3-72.

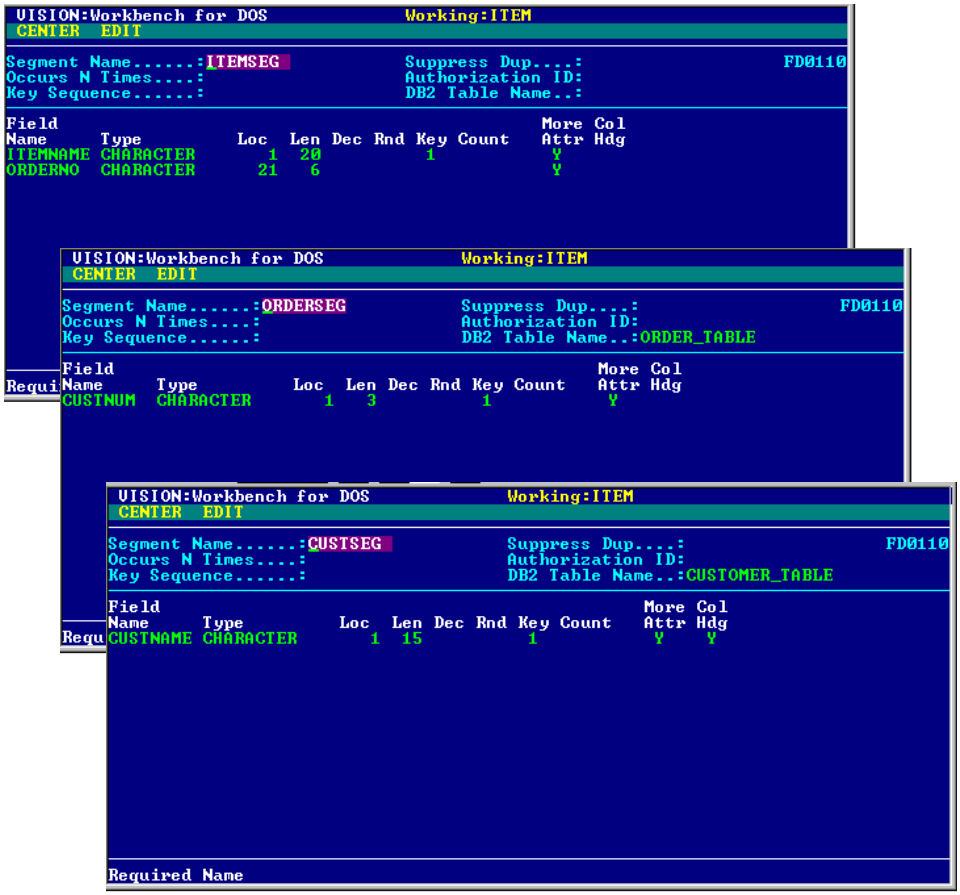



Figure 3-72 Example 4: Mapping Specification

It might be argued that this is not new, as a similar approach could be used with secondary indexing under IMS. This might be true for this example, but see Example 2 Creating an Organization Chart on page 3-59 for an example that cannot be achieved by secondary indexing.

## Example 5 Varying the View by Item

Example 4 Viewing the Customer-Order Database by Item on page 3-66 showed that by taking a different view, a series of item records could be constructed, each containing a customer. Treating the logical file as unsequenced enabled you to process the records with duplicate keys. This example constructs a logical record for each item that contains all of the customers who purchased the item.

This is done by constructing an item root segment that is independent of any specific order number and establishing all of the item/order combinations as lower level segments. It effectively gathers all of the duplicate records created in Example 4 Viewing the Customer-Order Database by Item on page 3-66 under the umbrella of a single root segment.

```
CUSTOMER_TABLE
CUSTOMER_NO, CUSTOMER_NAME, CUSTOMER_PHONE

ORDER_TABLE
CUSTOMER_NUM, ORDER_NUMBER, ORDER_DATE, PO_NUMBER, AGENT

ORDERED_ITEMS_2
ORDER_NO, ITEM_NAME, ITEM_PRICE, QUANTITY
```

Figure 3-73 Example 5: Table Information

The structure is shown in Figure 3-74.

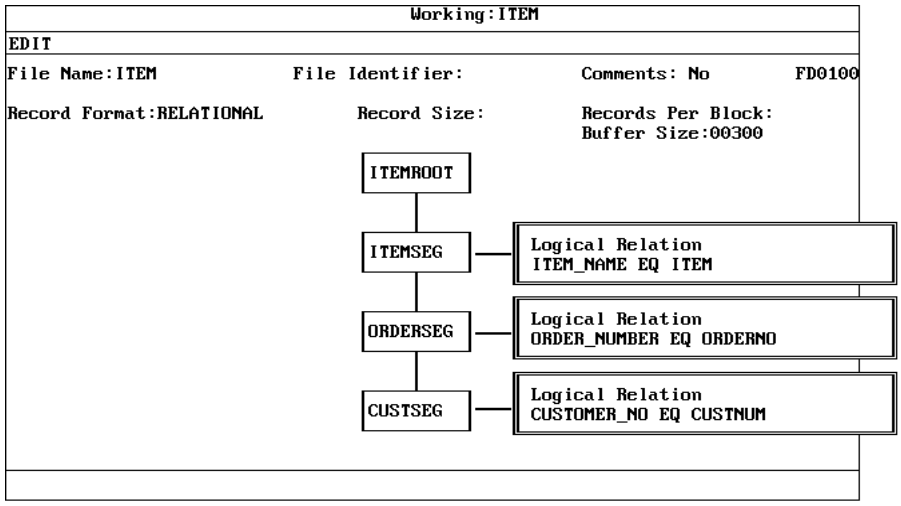



Figure 3-74 Example 5: Logical Structure

**Segment Name:** ITEMROOT

| Builder Field Name | DB2 Column Name |
|---|---|
| ITEM | ITEM_NAME |

**Segment Name:** ITEMSEG

| Builder Field Name | DB2 Column Name |
|---|---|
| ORDERNO | ORDER_No |

**Segment Name:** ORDERSEG

| Builder Field Name | DB2 Column Name |
|---|---|
| CUSTNUM | CUSTOMER_NUM |

**Segment Name:** CUSTSEG

| Builder Field Name | DB2 Column Name |
|---|---|
| CUSTNAME | CUSTOMER_NAME |

The mapping specification is shown in Figure 3-75.

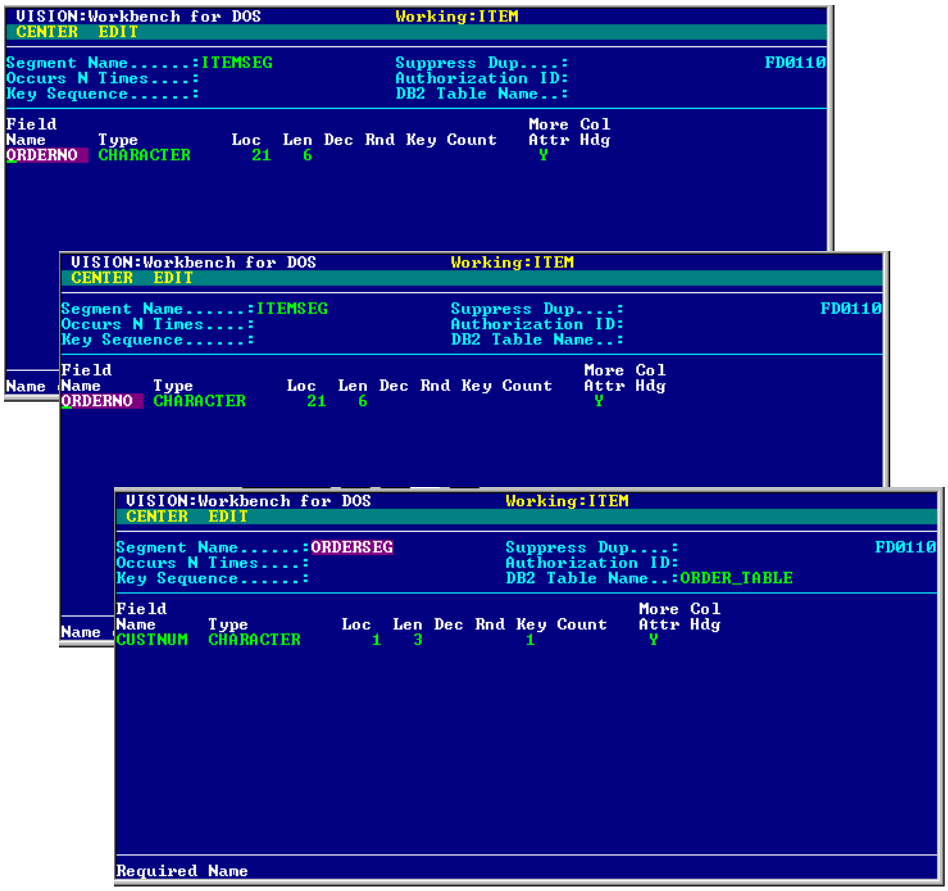



Figure 3-75 Example 5: Mapping Specification

This example shows that while each segment represents a table, there is no constraint that each segment represent a different table. Segments ITEMROOT and ITEMSEG both represent the same table (ORDERED_ITEMS_2).

As a root segment, ITEMROOT obtains the first row for an item as the root. However, ITEMSEG is governed by the logical relation that enforces the rule that all rows with the selected item name are brought into the record. In this case, that is all of the items with the same value as the previously obtained root.

This process includes retrieving a second copy of the row used for the root. Remember, level 2 is a different logical segment and its construction does not depend on I/O that was done for level 1. There are two reasons why you would want to construct an application this way.

- This method performs all data synchronization independent of any other processing. Additionally, it can be easier to process with all the data available in a single record.
- The suppress duplicate feature can be used to remove duplicate segments from the record automatically.

**Note:** These examples are provided for illustrative purposes and should be adapted to the context of the applications.

In this example, if you use it as described above, you would get several duplicate records. Each root segment creates a record. Therefore, you would get as many duplicate records as you have occurrences of items.

This technique, therefore, is best used when there is some method of identifying a single root segment (for example, using a transaction finder record or defining the data as an indexed coordinated file).

The reason that records with duplicate keys are generated in Example 4 is that the tables in Example 4 are not normalized. The technique shown in this example amounts to normalizing tables on the fly.

### Example 6 Attaching DB2 WHERE Clauses

VISION:Builder provides you with an extension for the support of relational databases. This is the WH statement. The WH statement provides you with the ability to attach DB2 WHERE clauses to SELECT statements. To use the WH statement, you must be familiar with DB2. Appendix C, *DB2 Reserved Words* has a list of DB2 reserved words.

The WH statement can be included in the application using the fixed format VISION:Builder syntax. It can also be entered in VISION:Workbench for ISPF on the WHERE panel (a panel that is automatically displayed when you specify DB2 on the FILES panel).

In this discussion, the WH statement will be shown using the fixed format syntax. See Chapter 16, *Database Access Methodology* for details on the WH statement in conjunction with how VISION:Builder constructs DB2 statements and see Appendix A, *Sample DB2 Applications* for details on including WH statements in an application.

When building a logical record, VISION:Builder opens a cursor and issues FETCH statements as appropriate. The cursor is governed by a SELECT statement that is created by VISION:Builder. The SELECT statement identifies the fields that are retrieved from the database together with any qualifications or restrictions. As you have seen, these qualifications and restrictions can be defined to VISION:Workbench for DOS as logical relations.

In practice, the logical relations are turned into WHERE clauses. Sometimes it is necessary to provide constraints or WHERE clauses that cannot be defined by the use of logical relations. Provisions to include such statements are made through the WH statement.

The WH statement, therefore, provides you with the ability to extend the WHERE clause as created by VISION:Builder by connecting any appropriate extension using the AND operator. This extension permits you to use constructs not supported by the logical relation (such as the LIKE verb) or to use more sophisticated qualifications such as subselects or even correlated subqueries. WHERE clause extensions can be provided for any segment in the logical record. They are not limited to the root segment.

Figure 3-76 illustrates the use of WH statements. They can be applied against any segment in any coordinated file or in the master file. In this example, the WH statements are applied against two segments in the master file, the DEPRTMNT segment and the EMPS segment. The logical record is qualified such that only the sales department (B01) root segment is available and, within that, only individuals who earn less than $30,000 and who have the letter S in their name are made available to the application.

```
------------------------------------------------------------------------------
   RUN     STMT   SEGMENT    FILE   WHERE
   NAME    TYPE    NAME      QUAL   PREDICATE
------------------------------------------------------------------------------
(DB2RUN1 ) (WH)   (DEPRTMNT)  (O)   (DEPTNO = 'B01'                             )
(DB2RUN1 ) (WH)   (EMPS    )  (O)   (NAME LIKE '%S%' AND SALARY < 30000          )
```

Figure 3-76 Example 6: WH Statements

The application would only see the records shown in Figure 3-77.

```
B01 SALES
      231      JONES      23      25000      L65
      444      ADAMS      23      25000      L65
```

Figure 3-77 Example 6: Logical Record

## Example 7 Selecting Segments with WH Statements, Suppressing Duplicate Segments, and Ordering Segments

The logical record concept involves constructing records that contain only the data of interest and only a portion of the data available. Even if the file definition reflects all of the data available in the table, VISION:Builder subsets the data to include only the fields that have been referenced or are needed for the application. This frequently generates segments where the data is identical. For some applications, it is useful to eliminate the duplicate data. An entry of YES in the Suppress Dup entry in the Segment Definition window achieves this.

Assume that you want to display the class and sex of all people who earn more than $20,000 per year. The structure of the logical record that achieves this is shown in Figure 3-78.

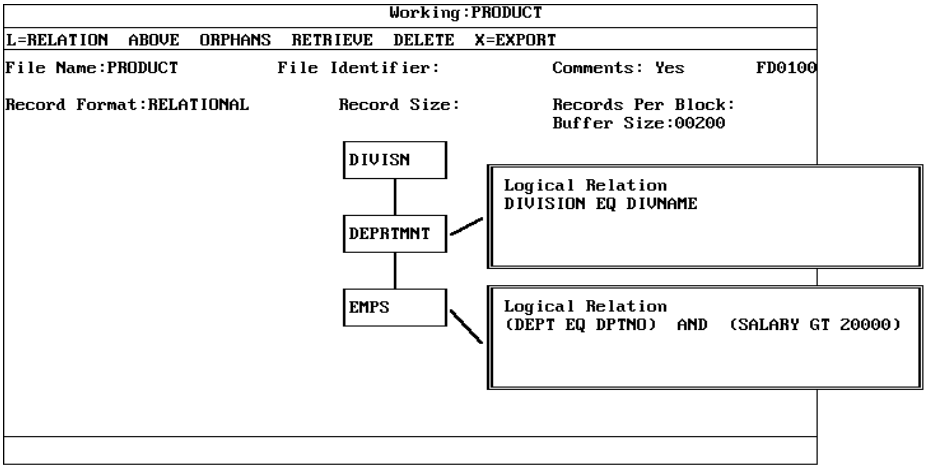



Figure 3-78 Example 7: Logical Structure

**Segment Name:** DIVISN

| Builder Field Name | DB2 Column Name |
|---|---|
| DIVNAME | DIV |

**Segment Name:** DEPRTNMT

| Builder Field Name | DB2 Column Name |
|---|---|
| DPTNO | DEPTNO |
| DEPTNAME | NAME |

**Segment Name:** EMPS

| Builder Field Name | DB2 Column Name |
|---|---|
| CLASS | CLASS |
| SEX | SEX |

The definition is shown in Figure 3-79.

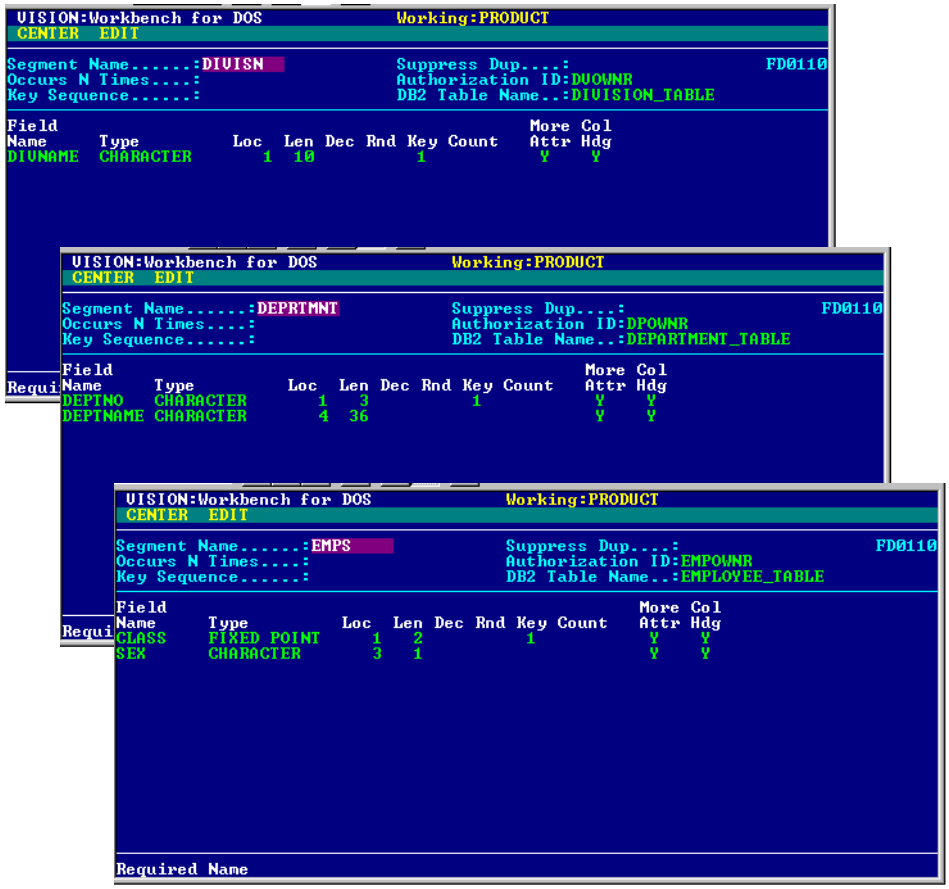



Figure 3-79 Example 7: Mapping Specification

This definition shows that logical relationships can consist of multiple expressions connected by the AND or OR operators. The rules can involve columns from the tables that are not in the logical records. In this example, DIVISION, DEPT, and SALARY are all fields that exist in the relational tables but are not included in the logical record. The example also illustrates that the rules can involve constants. The logical relation is the equivalent of the WHERE clause of a SELECT statement or, in the IMS world, logical relations serve as a function similar to SSAs.

Using the sample data, a typical record is shown in Figure 3-80.

```
WIDGETS
      A01 FINANCE
          45    M
          23    F
      E05 LEGAL
```

Figure 3-80 Example 7: Logical Record

```
B01 SALES
    23    M
    34    M
    29    F
    34    F
    23    M
C02 ENGINEERING
    44    M
    44    M
```

Figure 3-80 Example 7: Logical Record

When displayed on a report, you see duplicate line entries, just as in the record representation.

Notice the outer join feature of E05 LEGAL. Although this department has no employees, it is not lost from the record. A VISION:Builder report will correctly reflect that there are no applicable employees in this department; it will not omit a department that has no data satisfying the criteria.

Had you modified the definition to request the suppression of duplicate data, the logical structure, as shown in Figure 3-81, would be unchanged from the one in Figure 3-78.

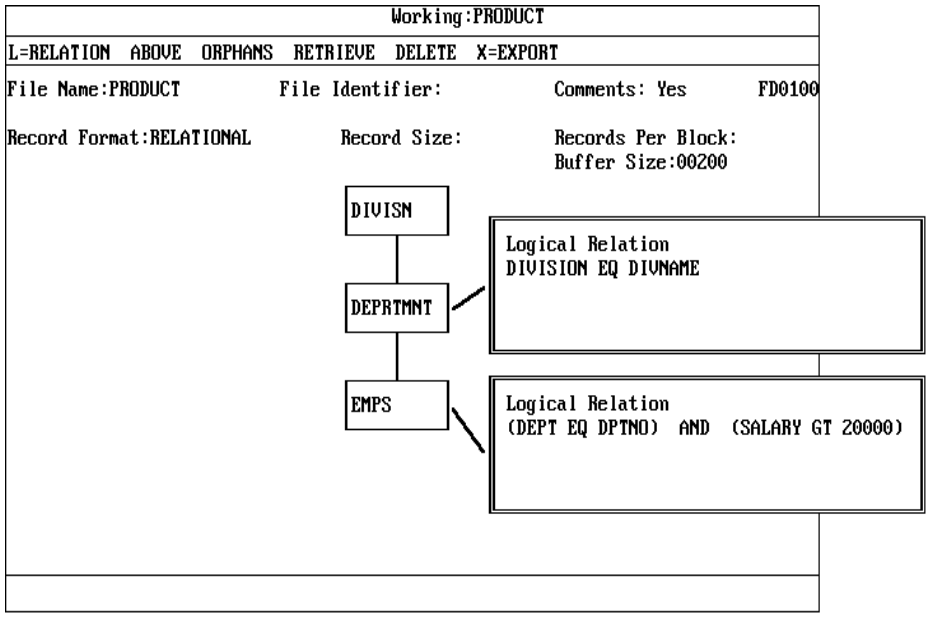



Figure 3-81 Example 7: Logical Structure with Suppression of Duplicate Data

**Segment Name:** DIVISN

| Builder Field Name | DB2 Column Name |
|---|---|
| DIVNAME | DIV |

**Segment Name:** DEPRTNMT

| Builder Field Name | DB2 Column Name |
|---|---|
| DPTNO | DEPTNO |
| DEPTNAME | NAME |

**Segment Name:** EMPS

| Builder Field Name | DB2 Column Name |
|---|---|
| CLASS | CLASS |
| SEX | SEX |

The mapping specification is shown in Figure 3-82. The only change is the YES in the Suppress Dup entry.

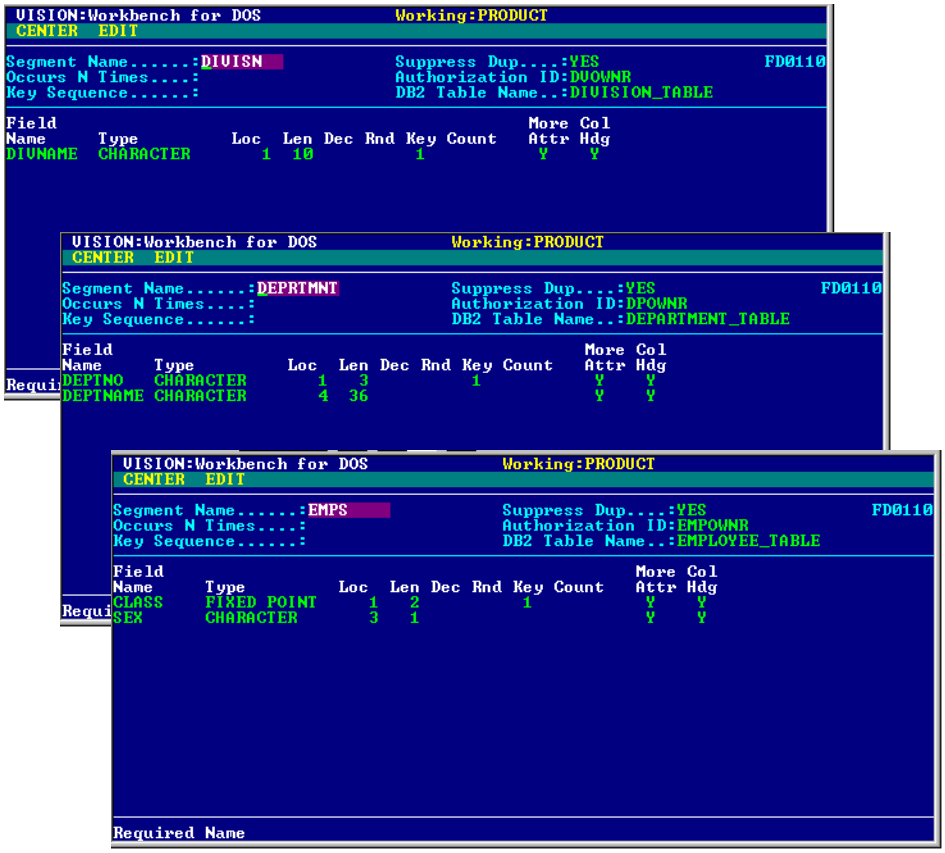



Figure 3-82 Example 7: Mapping Specification with Suppression of Duplicate Data

Using the sample data, a typical record with suppression of duplicate data is in Figure 3-83.

```
WIDGETS
     A01  FINANCE
          45    M
          23    F
     E05  LEGAL
     B01  SALES
          23    M
          29    F
          34    F
          34    M
     C02  ENGINEERING
          44    M
```

Figure 3-83 Example 7: Logical Record with Suppression of Duplicate Data

**Note:** DB2 does not guarantee that the suppression of duplicates will also provide a sorted result. There can be circumstances where a sorted order is not returned. If the segments are wanted in a sorted order, this should be explicitly identified in the Key Sequence entry of the Segment Definition window as illustrated below.

Duplicate segments are not included in the logical record. Notice that a segment is included in the logical record if any field is unique. Duplicate keys alone do not cause the suppression of duplicate segments. This illustration shows that the order of the rows in the logical record is different. This is because the Suppress Duplicates action causes a sort (by DB2) that results in the elimination of duplicates and presentation of the rows in a different order.

Extend the example to illustrate the capability of ordering the segments in the record. Segments (including root segments) can be defined to be unordered, in ascending order, or in descending order. This specification can be combined with the other specifications that you have used. The ordering specification is made in the Key Sequence entry of the Segment Definition window.

**Segment Name:** DIVISN

| Builder Field Name | DB2 Column Name |
|---|---|
| DIVNAME | DIV |

**Segment Name:** DEPRTNMT

| Builder Field Name | DB2 Column Name |
|---|---|
| DPTNO | DEPTNO |
| DEPTNAME | NAME |

**Segment Name:** EMPS

| Builder Field Name | DB2 Column Name |
|---|---|
| CLASS | CLASS |
| SEX | SEX |

The mapping specification in Figure 3-84 shows a key sequence entry of DESCENDING for the DEPRTMNT segment and ASCENDING for the EMPS segment.

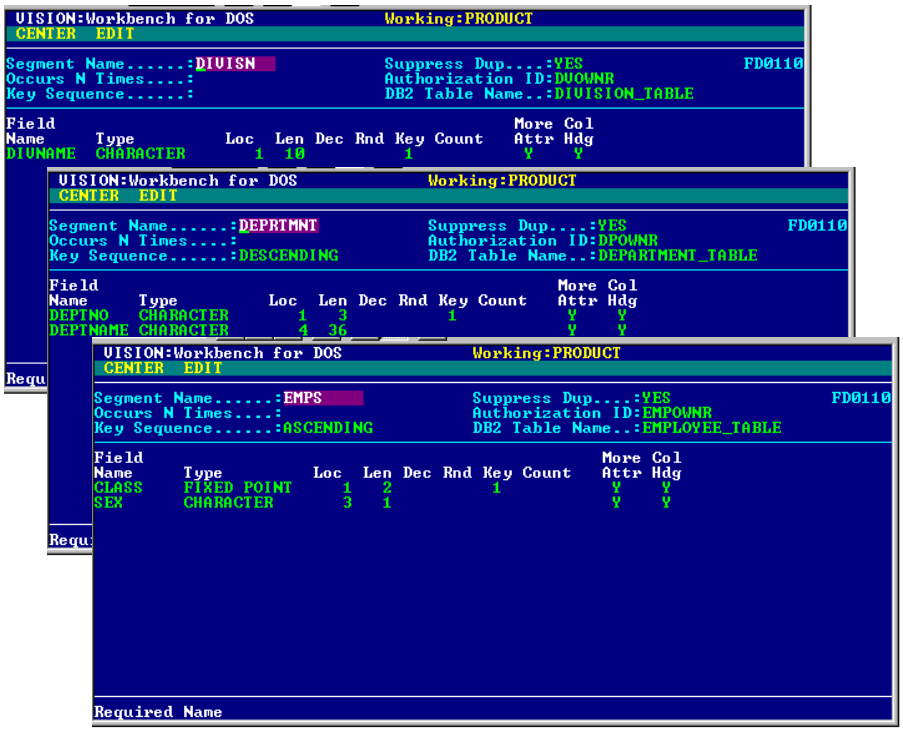



Figure 3-84 Example 7: Mapping Specification with Suppression of Duplicate Data and Ordering of Data

The record now can be visualized as shown in Figure 3-85.

```
WIDGETS
     E05  LEGAL
     C02  ENGINEERING
          44    M
     B01  SALES
          23    M
          29    F
          34    F
          34    M
     A01  FINANCE
          23    F
          45    M
```

Figure 3-85 Example 7: Logical Record with Suppression of Duplicate Data and Ordering of Data

The capability of ordering segments can be very useful in designing well organized production applications. Data presented in alphabetical or numerical order can greatly facilitate processing.

There is one caveat here. The ordering of segments and the suppression of duplicate segments are actually performed by DB2 per VISION:Builder's instructions. Achieving these results requires the database manager to consume additional CPU cycles. This could be costly if there are no DB2 indexes on the fields concerned. Creating indexes on the tables is a database manager function external to VISION:Builder.

Up to now, the focus has been on the logical record, but it would be unusual to code a constant value for salary (in the logical relation) in the definition of a logical record. This is much more of a run time parameter than a file definition parameter. Run time parameters are provided by the WH statement. The following will show how this problem can be implemented with a combination of file definition (logical relation) and run time parameters (WH statements).

Remove the logical relation that references SALARY from the previous file definition and include that specification on the WH statement. The remaining mapping specifications and field definitions remain the same.

The logical structure is shown in Figure 3-86. Note that the structure itself is unchanged; only the logical relationship is changed.

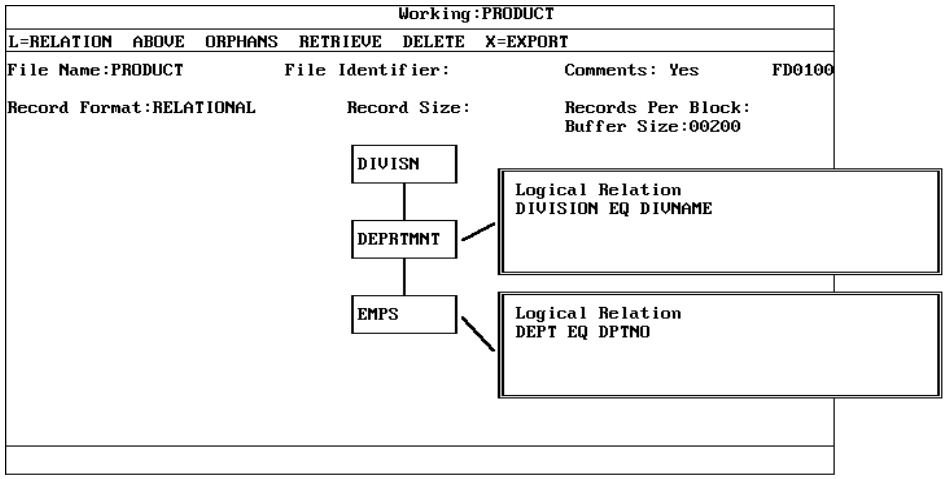



Figure 3-86 Example 7: Mapping Specifications with a WH Statement

The WH statement needed at run time is shown in Figure 3-87.

```
-------------------------------------------------------------------------------------
   RUN    STMT   SEGMENT    FILE   WHERE
   NAME   TYPE    NAME      QUAL   PREDICATE
-------------------------------------------------------------------------------------
(DB2RUN1 ) (WH)   (EMPS    )  (O)   (SALARY > 20000                                )
```

Figure 3-87 Example 7: Source Statements with a WH Statement

Using the sample data, a typical record is shown in Figure 3-88. Note that it is no different than the record previously shown in this example; it was just achieved with different specifications.

```
WIDGETS
     E05  LEGAL
     C02  ENGINEERING
          44    M
     B01  SALES
          23    M
          29    F
          34    F
          34    M
     A01  FINANCE
          23    F
          45    M
```

Figure 3-88 Example 7: Logical Record with a WH Statement

## Example 8 Tables with Hierarchically Organized Keys

This example illustrates defining a logical record where the keys are in a strictly hierarchical organization.

Assume you have relational tables as shown in Figure 3-89.

```
CUSTOMER_TABLE
CUSTOMER_NO, CUSTOMER_NAME, CUSTOMER_PHONE

ORDER_TABLE
CUSTOMER_NUM, ORDER_NUMBER, ORDER_DATE, PO_NUMBER, AGENT

ORDERED_ITEMS
CUSTOMER_NUMBER, ORDER_NO, ITEM_NAME, ITEM_PRICE, QUANTITY
```

Figure 3-89 Example 8: Relational Tables

The relational column names are chosen to be different for clarity. In practice, because these are separate tables, it will be logical to use the same names for like columns. For example, the customer number would probably be CUSTOMER_NUMBER in all three tables.

The logical structure is shown in Figure 3-90.

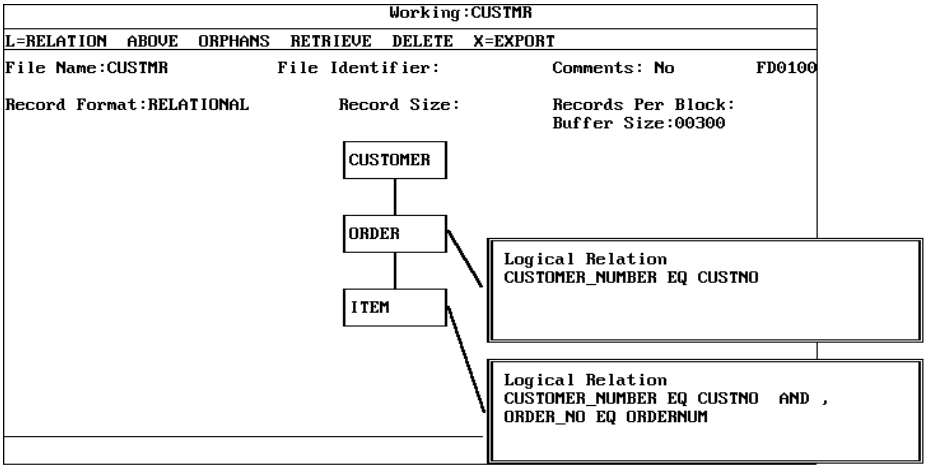



Figure 3-90 Example 8: Logical Structure

**Segment Name:** CUSTOMER

| Builder Field Name | DB2 Column Name |
|---|---|
| CUSTNO | CUSTOMER_NO |
| CUSTNAME | CUSTOMER_NAME |
| CUSPHONE | CUSTOMER_PHONE |

**Segment Name:** ORDER

| Builder Field Name | DB2 Column Name |
|---|---|
| ORDERNUM | ORDER_NUMBER |
| ORDATE | ORDER_DATE |
| PONUM | PO_NUMBER |
| AGNT | AGENT |

**Segment Name:** ITEM

| Builder Field Name | DB2 Column Name |
|---|---|
| ITEMNAME | ITEM_NAME |
| PRICE | ITEM_PRICE |
| ITEMQTY | QUANTITY |

The mapping specification is shown in Figure 3-91.

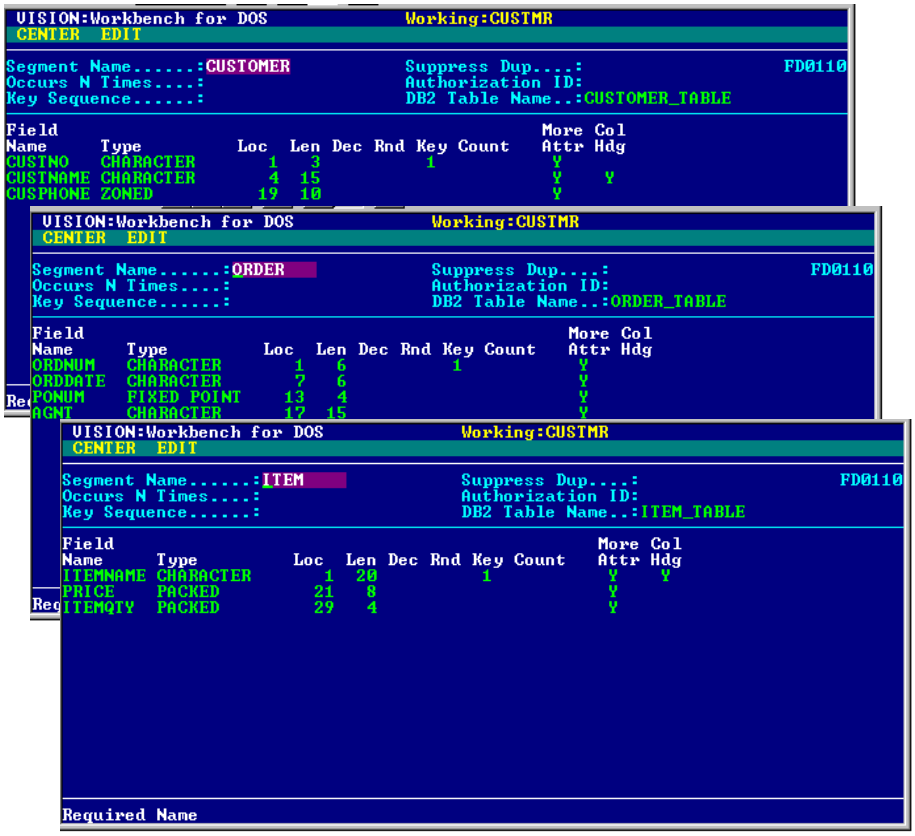



Figure 3-91 Example 8: Mapping Specification

Summarizing briefly, the file and segment definitions are defined in the usual way but augmented to add the relational names. The Authorization ID is not shown, implying that all tables have the same Authorization ID. This can be provided on the Files Used window (or in fixed format syntax on the RF statement at execution time) if it is different from the sign-on user ID. Otherwise, DB2 uses the sign-on user ID.

The logical relations are included to make sure that the right subordinate segments are attached to the corresponding parents. Notice in the logical relation for the ITEM segment that the segments are attached to the correct ORDER *within* CUSTOMER and that the VISION:Builder field names apply to higher level segments but not necessarily the immediate parent (CUSTNO is in the root segment).

## Example 9 Relational Tables Where the Keys are not in Hierarchical Order

Assume that order numbers are unique across all customers. In this case, the table of ordered items does not contain the customer number (knowing the order number is sufficient to establish where the ordered items go). You are also only interested in customers with numbers in the range of 100 to 200 or in the range of 500 to 600.

The tables are shown in Figure 3-92.

```
CUSTOMER_TABLE
CUSTOMER_NO, CUSTOMER_NAME, CUSTOMER_PHONE

ORDER_TABLE
CUSTOMER_NUM, ORDER_NUMBER, ORDER_DATE, PO_NUMBER, AGENT

ORDERED_ITEMS_2
ORDER_NO, ITEM_NAME, ITEM_PRICE, QUANTITY
```

Figure 3-92 Example 9: Relational Tables

The logical structure is shown in Figure 3-93.

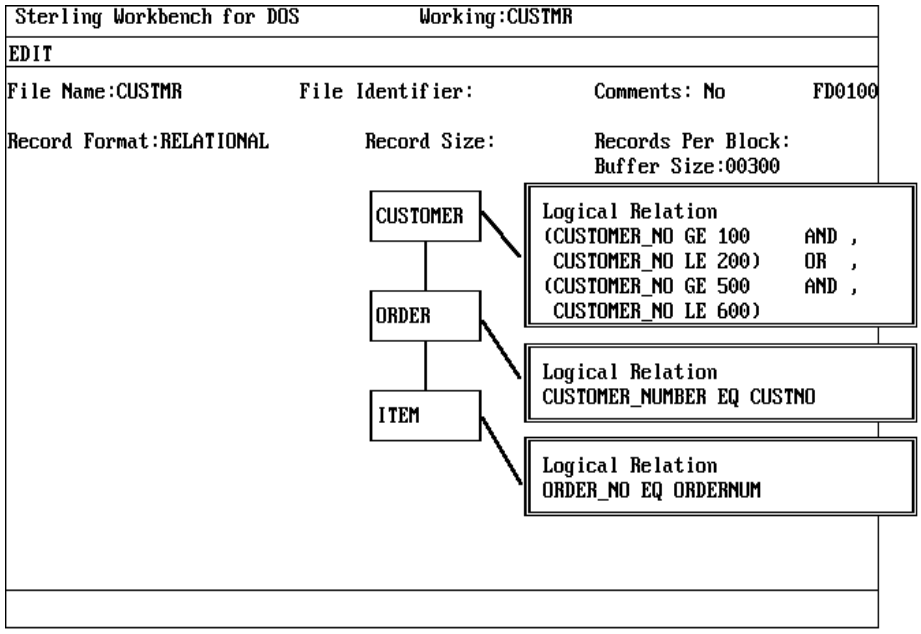



Figure 3-93 Example 9: Logical Structure

**Segment Name:** CUSTOMER

| Builder Field Name | DB2 Column Name |
|---|---|
| CUSTNO | CUSTOMER_NO |
| CUSTNAME | CUSTOMER_NAME |
| CUSPHONE | CUSTOMER_PHONE |

**Segment Name:** ORDER

| Builder Field Name | DB2 Column Name |
|---|---|
| ORDERNUM | ORDER_NUMBER |
| ORDATE | ORDER_DATE |
| PONUM | PO_NUMBER |
| AGNT | AGENT |

**Segment Name:** ITEM

| Builder Field Name | DB2 Column Name |
|---|---|
| ITEMNAME | ITEM_NAME |
| PRICE | ITEM_PRICE |
| ITEMQTY | QUANTITY |

The mapping specification is shown in Figure 3-94.

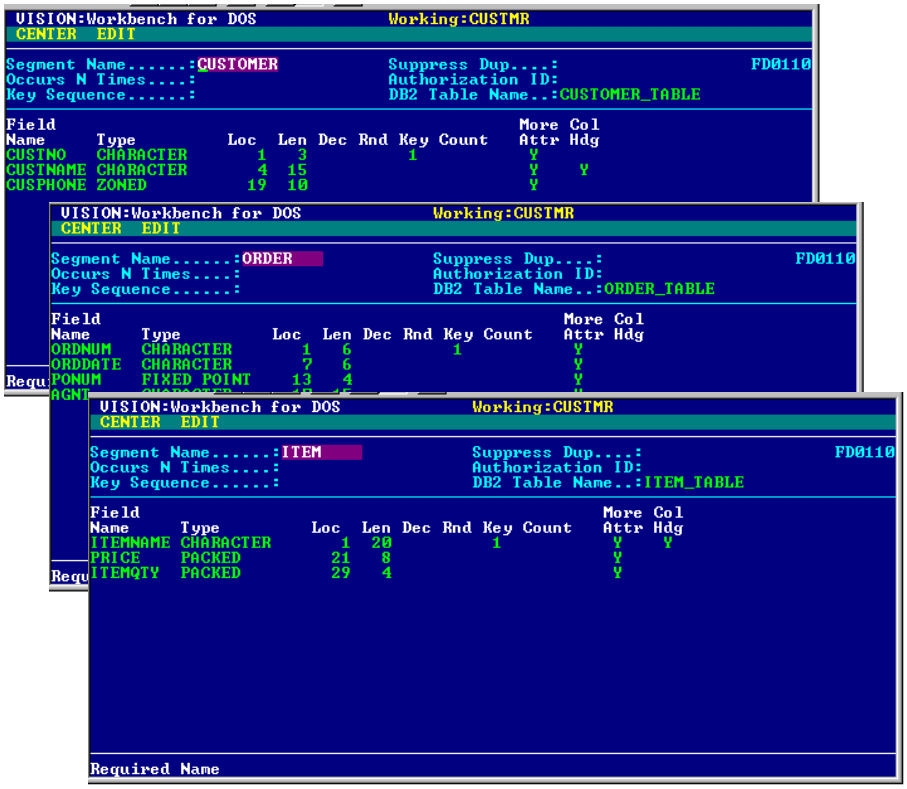



Figure 3-94 Example 9: Mapping Specification

Logical relations can also be applied to the root segment, thereby limiting the application's view to only those logical records of interest. Establishing the parentage of the ITEM segment requires only reference to its immediate parent. This example shows that the relational tables do not have to have a series of fields in a strict hierarchy to generate a logical record.

The logical relation on the root segment is not used to establish a relationship but to filter the available records. This same filter could have been achieved by a procedure in the application. This would, however, have cluttered up the understanding of the application. It also would have made the application more inefficient in terms of CPU resources because every row would be required to be passed from DB2 to the application, which would then perform the filtering.

The filtering can also be achieved by using WH statements, included in the application (run time) rather than the file definition. By removing the logical relation from the CUSTOMER root segment in the file definition and including two similar WH statements in the application, the filtering criteria becomes a run-time specification (see Figure 3-95).

```
------------------------------------------------------------------------------------------
   RUN     STMT   SEGMENT     FILE   WHERE
   NAME    TYPE    NAME       QUAL   PREDICATE
------------------------------------------------------------------------------------------
(DB2RUN1 ) (WH)   (CUSTOMER)  (O)    (CUSTOMER_NO BETWEEN 100 AND 200  OR             )
(DB2RUN1 ) (WH)   (CUSTOMER)  (O)    (CUSTOMER_NO BETWEEN 500 AND 600                 )
```

Figure 3-95 Example 9: WH Statements

## Example 10 Using a Column Function

This example illustrates the use of DB2 column functions: AVG, COUNT, MAX, MIN, and SUM. Suppose you want the sum of all the salaries in each department. At first, you can be inclined to build each logical record and request a VISION:Builder total on SUMSAL with DEPT as the control break.

But if you think of the data as how you would like it to be and not as how it physically exists, you can utilize the power of the DB2 column function SUM. This can be accomplished with just a change in the file definition.

**Segment Name:** EMPS

| Builder Field Name | DB2 Column Name |
|---|---|
| DEPT | DEPT |
| SUMSAL | SUM (SALARY) |

The logical structure and mapping specifications used to achieve this are shown in Figure 3-96.

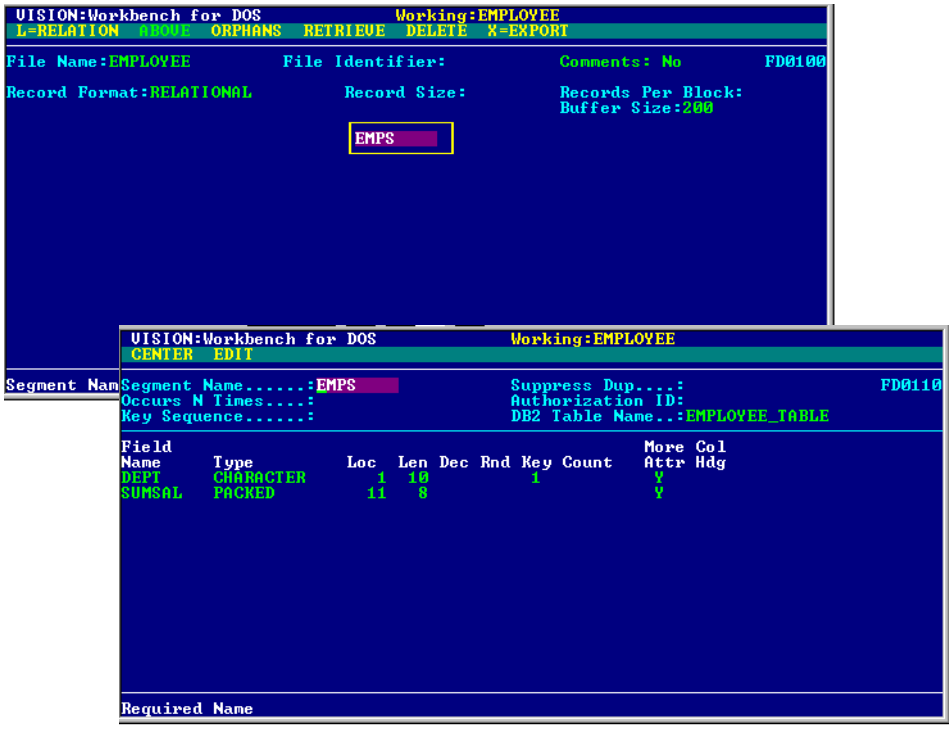



Figure 3-96 Example 10: Logical Structure and Mapping Specification

Using the sample data, a typical record is shown in Figure 3-97.

```
A01   $100,000
E05       -
B01   $175,000
C02   $ 92,000
```

Figure 3-97 Example 10: Logical Record

This has eliminated almost all of the overhead of building the logical record and has entirely freed the application of the task of computing the total of the salaries. The application needs only to report the two fields. The database manager has done all the work.

The DB2 SELECT statement that VISION:Builder generates consists of all referenced fields and all key fields, plus a GROUP BY clause (because a column function is involved). The GROUP BY clause contains all selected fields that are not column functions (see Figure 3-98).

```
SELECT DEPT, SUM(SALARY) GROUP BY DEPT
```

Figure 3-98 Example 10: Generated SELECT Statement

Because key fields are included in the SELECT statement, picking a unique key field for the segment will, in fact, nullify the effect of the GROUP BY. Therefore, the key field chosen for the segment should not uniquely identify the segment.

Also, because DB2 does not allow a column function on an UPDATE statement, any file containing a column function cannot be updated by VISION:Builder.

### Example 11 Defining a Multi-Column Keyed Segment

It frequently requires more than one column of a relational table to uniquely identify a row. Suppose you have a table as shown in Figure 3-99 that contains the detailed information about each department (for example, the division and department name and the city in which it is located). The physical data is shown in Figure 3-100. To uniquely identify a row requires both the division number and the department number, because the same department numbers are used within each division.

```
@REPORT TEXT =  DEPTINFO_TABLE                                          <R>
                                                          <R>
 DIVISION_NUM, DEPT_NUM, DIVISION_NAME, DEPT_NAME, DEPT_LOC
```

Figure 3-99 Example 11: Logical Structure

```
@REPORT TEXT =  DEPTINFO_TABLE                                                  <R>
                                                                  <R>
 DIVNO     DEPTNO     DIVNAME         DEPTNAME         DEPTLOC   <R>
                                                                  <R>
 W20000    A01        SYSTEMS         FINANCE          HOUSTON   <R>
 W20000    A04        SYSTEMS         SALES            DENVER    <R>
 W20000    A26        SYSTEMS         ENGINEERING      HOUSTON   <R>
 R88000    A01        INSURANCE       FINANCE          DALLAS    <R>
 R88000    A04        INSURANCE       SALES            DALLAS    <R>
 R88000    A22        INSURANCE       MARKETING        NEW YORK  <R>
 R88000    A26        INSURANCE       ENGINEERING      HOUSTON
```

Figure 3-100 Example 11: Table Information

When VISION:Builder directly reads a segment, it uses only one key field — the primary key field. Any additional key fields that can be defined are ignored. However, one key may not be enough to retrieve a unique row. For segments in a relational file that are accessed directly by VISION:Builder and require more than one column to identify a unique row, you can define the segment as a multi-column keyed segment to VISION:Builder. This effectively accesses rows based on multiple column values and retrieves a unique row.

In a multi-column keyed segment, you define a primary key that is composed of multiple columns from the relational table. This set of columns is used by VISION:Builder when performing a direct-read of the segment, effectively accessing rows based on multiple column values. You are also required to supply supplemental WH statements in any application that accesses these segments. Chapter 16, *Database Access Methodology* contains more information concerning the circumstances under which you can use multi-column keyed segments. This section instructs you on how to define them and how to construct the necessary WH statements.

## Defining the Primary Key Field

When defining a multi-column keyed segment in a relational file definition, the primary key is a character field defined without a relational column name. The primary key is overdefined by any number of fields, each having a corresponding relational column name. These component fields specify the set of columns that comprise the primary key and their total length is equal to the length of the

primary key. Each character of the primary key must be overdefined. There can be only one key field defined. Any additional key fields specified are accepted but will be ignored during processing.

The Segment Definition window in Figure 3-101 shows the DEPRTMNT segment from the DEPTINFO_TABLE (see Figure 3-99). It is defined as a multi-column keyed segment. The primary key, KEYFIELD, is defined by entering 1 for Key on the Segment Definition window. The Type is defined as Character. The Length entry of 9 is the sum of the lengths of the two fields that it overdefines — DIVNO and DEPTNO. The DB2 Name entry for the primary key is left blank. Note that each character of the primary key is overdefined, as required.

**Segment Name:** DEPRTMNT

| Builder Field Name | DB2 Column Name |
|---|---|
| DIVNO | DIVISION_NUM |
| DEPTNO | DEPARTMENT_NUM |

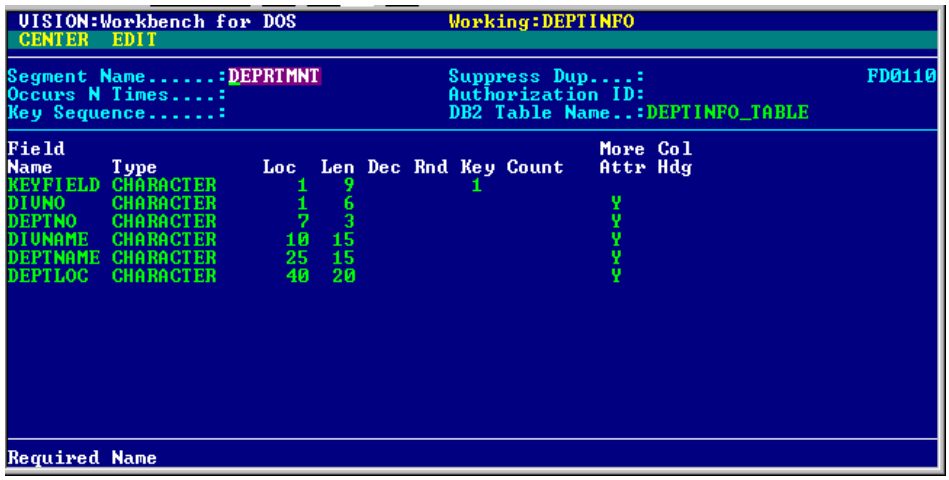



Figure 3-101 Example 11: Segment Definition Window for DEPTNO Segment

The component fields, DIVNO and DEPTNO, are defined with Location and Length entries that overdefine the entire KEYFIELD field without any gaps. The Key entries are left blank. The required DB2 column names are provided for each of these component fields. This set of columns, DIVISION_NUM and DEPARTMENT_NUM, identifies a unique row in the table.

Although component fields can be of any field type and length, special attention should be given to these attributes. Because they make up a composite key that is defined as character, any comparison done by VISION:Builder on the primary key will be a string compare, not a numeric compare. A string compare of two numeric fields can yield incorrect results.

## Defining the Match Field

The match field is the field that contains the actual key value that controls the direct-read. The match field must be defined according to the same rules as the primary key. It must be a character field of the same length as the primary key of the multi-column segment. It must be overdefined by the host variables coded on the WH statements. This is necessary because VISION:Builder is basing its coordination on just one key field and one coordinating field. Any comparison done to determine if the coordination is successful or if you must coordinate again is based on this single field.

## Constructing the Supplemental WH Statements

Any application that accesses a multi-column keyed segment must also include supplemental WH statements for the segment. WH statements are not supported by VISION:Workbench for DOS. See Appendix B, *VISION:Builder System Component Relationships* for details on including WH statements in an application.

VISION:Builder includes the information from the supplemental WH statements at the end of the WHERE clause of the SELECT statement it generates to retrieve a row from the table by key. The WHERE clause contains the logical relationship from the file definition, if any, followed by any WH statements you supply.

The WH statements must include a comparison of each relational column (from the component fields that overdefine the primary key of the segment) to a portion of the match field, coded as a host variable. The match field is the field that contains the actual key value that controls the direct-read.

## Example of a Multi-Column Keyed Segment

For the purpose of this discussion, consider the most common instance that requires a direct-read of a segment — an indexed coordinated file. VISION:Builder directly reads the root segment of an indexed coordinated file by key. It matches the primary key of the indexed coordinated file to any coordinating field you specify. See Chapter 6, *Input File Processing* for more information on indexed coordinated files.

Suppose you have an application that uses the EMPLOYEE definition (in Figure 3-102) as the master file and the DEPTINFO definition (in Figure 3-101) as an indexed coordinated file. You want to report each employee and the name of the division and department in which the employee works. This requires that both the division number and department number from the master file be used as key values to retrieve a unique row from the DEPTINFO table.

First, you define the root segment of the indexed coordinated file as a multi-column keyed segment (see Figure 3-101). The primary key field, KEYFIELD, is overdefined, as required, by the two fields, DIVNO and DEPTNO, whose related columns make a row unique.

Next, you specify a match field and overdefine it as required. For an indexed coordinated file, the match field is the coordinating field. The field name of the coordinating field is entered on the Files Used window in the Alternate Key(s) entry. The file qualifier (indicating the work area containing the coordinating field) is entered in the ICF entry of the Files Used window.

The Files Used window in Figure 3-103 on page 3-89 shows that the MATCHFLD field from the master file is the coordinating field for M4CORD1. The MATCHFLD field is overdefined, as required, by the DIVISION and DEPRTMNT fields (see Figure 3-102).

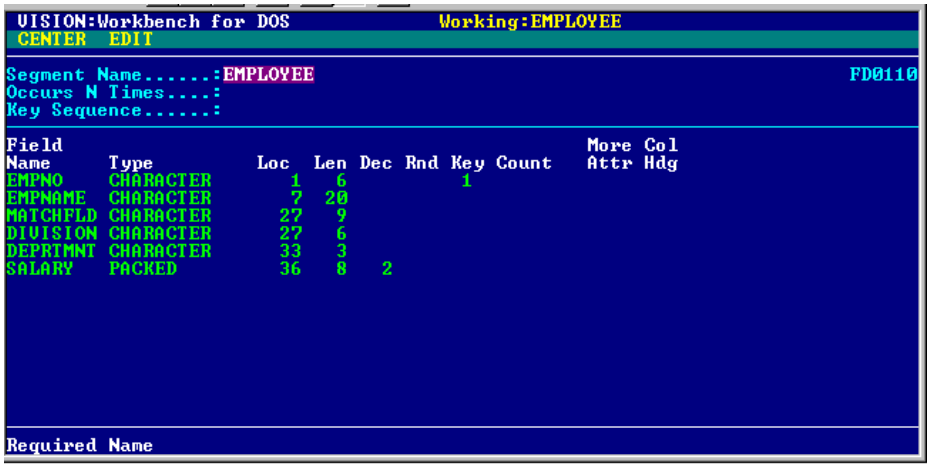



Figure 3-102 Example 11: Segment Definition Window for EMPLOYEE Segment

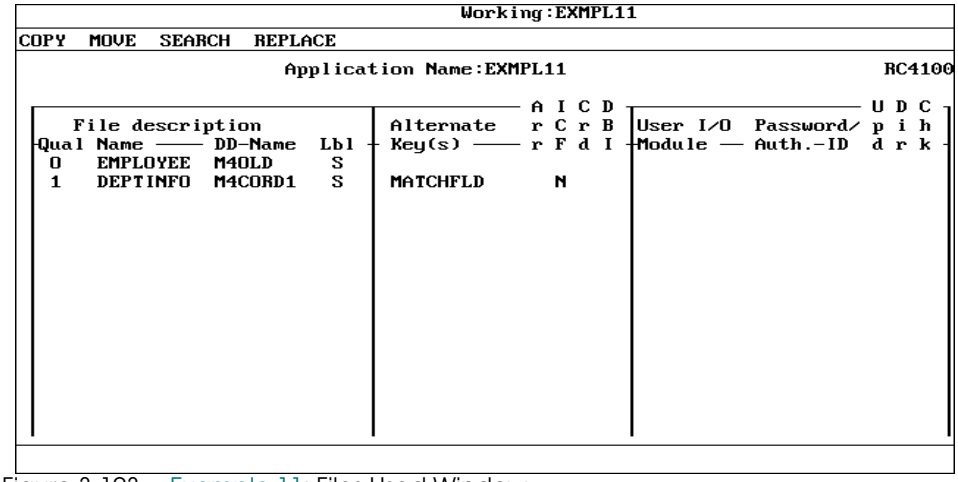



Figure 3-103 Example 11: Files Used Window

Now, you construct the supplemental WH statements required in this application. They must compare each component of the primary key field to a host variable that overdefines a portion of the coordinating field. Figure 3-104 shows the WH statements.

```
@REPORT TEXT = ------------------------------------------------------------------------------------------<R>
   RUN    STMT   SEGMENT    FILE   WHERE<N>                                                    <R>
   NAME   TYPE    NAME      QUAL   PREDICATE                                                 <R>
-------------------------------------------------------------------------------------------<R>
(RUN001  ) (WH)   (ROOTSEG )  (1)   (DIVISION_NUM    = :N.DIVISION   AND          )     <R>
(RUN001  ) (WH)   (ROOTSEG )  (1)   (DEPARTMENT_NUM = :N.DEPARTMENT              )
```

Figure 3-104 Example 11: WH Statements

During the application, whenever the indexed coordinated file is to be read, the row that matches the current values of the two host variables is retrieved.

The diagram in Figure 3-105 is helpful in understanding the fundamentals of defining the necessary items for a multi-column keyed segment.

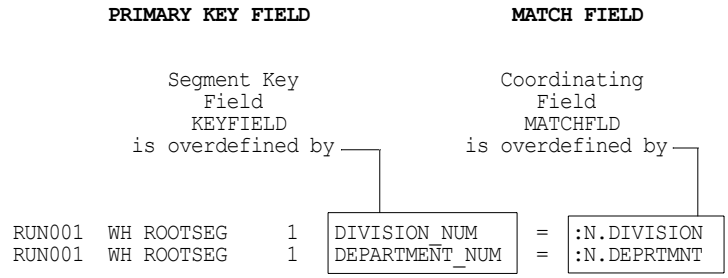



Figure 3-105 Example 11: Multi-Column Keyed Segments

# GDBI Files

The topic of GDBI (mapped) files is described in the GDBI Programming Manual.

# Compressed Files

To help reduce physical storage requirements, VISION:Builder provides a special compressed (packed) record format. When a file is defined as packed, the records are compressed by VISION:Builder before they are stored on a data device and expanded by VISION:Builder when accessed. The compression is performed without regard to field boundaries or field type and depends only on the data, not the file definition.

Compression of records is obtained by deleting certain repeating characters and replacing them with a special one-byte control field. A string of repeating characters can be compressed when it consists of three or more identical characters. The following character types are recognized:

- X'40' Character string blank
- X'F0' Zoned decimal zero
- X'00' Packed/binary zero

A file containing compressed records is defined to VISION:Builder by specifying Packed (type P) in the Record Format entry on the File or Transaction Definition windows. Packed records interface with the operating system as variable length records. Buffer size or record size is defined in the same manner as variable length records. As with variable length records, 12 is the minimum record size allowed for packed records.

Packed subfiles can also be defined. In this manner, an existing file can be converted to packed records in a single VISION:Builder run.

The manner in which records are compressed is transparent. When a compressed record is accessed, it is expanded (decompressed) and made available for processing exactly as described in the file definition.

At the conclusion of a VISION:Builder run, a compression factor is computed and printed with other statistics relevant to the file. The compression factor is computed as follows:

$$F = \frac{\Sigma\ (\text{in memory record size}) - (\text{compressed record size})}{\Sigma\ (\text{in memory record size})} \times 100$$

The size of the compression factor indicates the efficiency of the record compression: the larger the number, the greater the compression. For example, a compression factor of 40 indicates that the file takes 40 percent less space on the physical device after compression.

Figure 3-106 shows a detailed description of a packed record.

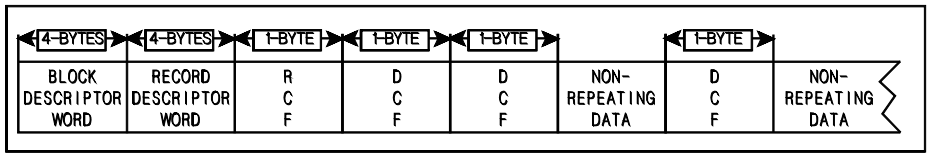



Figure 3-106 Sample Packed Record

| | | |
|---|---|---|
| Descriptor Word | Block | Length of the physical record. |
| | Record | Length of the subsequent logical record. |

| | |
|---|---|
| RCF (record control field) | A 1-byte field that is the first byte of every logical record.<br>Bit 0 = 1 Record has been compressed.<br>Bit 0 = 0 No compression in this record.<br>Bits 1-7 Reserved for VISION:Builder use. |
| DCF (data control field) | Contains the length of the subsequent non-repeating characters or the character type and length of the character string. A detailed illustration of the DCF is shown in Figure 3-107. |
| Non-Repeating Data | A character string that has no compression characters repeated for more than two consecutive bytes. |

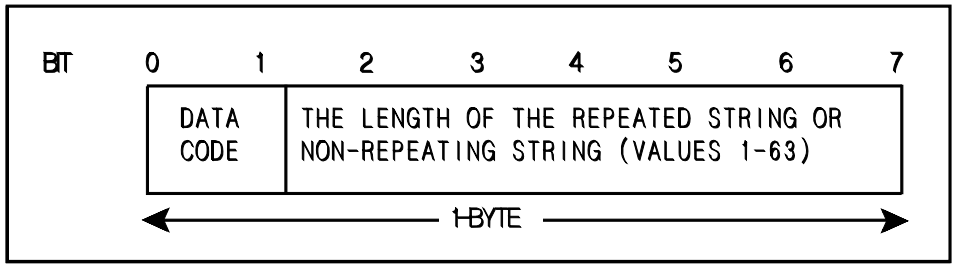



Figure 3-107 Data Control Field

| Data Code | Description |
|---|---|
| B'00' | The following data string consists of non-repeating characters. The data length is the number of characters in the string. |
| B'01' | This DCF replaces a string of repeating blank characters (X'40'). The data length is the number of repeated characters that existed prior to compression. |
| B'10' | This DCF replaces a string of repeating zoned decimal zeros (X'F0'). The data length is the number of repeated characters that existed prior to compression. |
| B'11' | This DCF replaces a string of binary/packed zeros (X'00'). The data length is the number of repeated characters that existed prior to compression. |

# Working Storage

Working storage refers to an alternate form of temporary storage provided by VISION:Builder. It is defined with one or more, up to 99, VISION:Builder file definitions and can be viewed as a single-record, single-level file resident in main storage. Fields in working storage are initialized to their null (empty) values at the beginning of processing and other data is placed in them only as a result of user action. Working storage is especially convenient as an area to hold data for communication with called routines.

Because the fields in working storage are modified only when you move data into them, you can be sure of their contents, regardless of the VISION:Builder automatic processing. Because the space is provided for temporary storage of data, you can also be sure you are not affecting or altering file information or VISION:Builder control fields. You can specify overdefinition of fields and up to nine column heading lines. It also allows you to predefine all temporary fields to be used in a run, as long as they use the working storage qualifier W and do not have an initial value other than blank or zero.

All of the definitions associated with working storage must be single level. Fixed length record format can be specified in a working storage definition. Because several definitions can be used, field names must be unique across all definitions. In a VISION:Builder procedure, the qualifier W is used to reference a field in working storage.

A working storage file definition must adhere to the following rules:

- The definition must describe an unstructured (single level) file.
- The definition cannot include variable length fields.
- When more than one file definition is used to describe the working storage section, all fields must be uniquely named across all definitions. Because all fields in working storage are qualified with W, the unique identification of a field must come from its defined name.
- Each working storage definition must be entered on the VISION:Workbench for DOS Files Used window of the application using it with a ddname entry of M4WORKnn (where nn is a number between 01 and 99).
- The total length of all working storage definitions (M4WORK01 through M4WORK99) must not exceed 32,767 bytes.

# Sample Definition/Maintenance Run

Chapter 2, *Application Development System* introduced the concept of a definition/maintenance run in which file definitions, and so on, are cataloged in the common library. Figure 3-108 is an example of a listing for a definition/maintenance run for VISION:Builder using a very simple file definition. Print lines in the listing with numbers on the right hand edge of the

listing represent the source statements of the definition/maintenance run. Lines without numbers on the right hand edge are heading and message lines inserted by VISION:Builder for readability.

The first statement of any VISION:Builder run must be a Run Control (RC) statement that specifies the input and output files for an application and other run parameters. In most instances, the RC statement for a definition/maintenance run is left entirely blank except for the Run Name and the Statement Type. If you are not using VISION:Workbench for DOS for preparing your definitions, refer to the *VISION:Builder Reference Summary* for a detailed description of the syntax for each statement type. When using VISION:Workbench for DOS, the RC statement is supplied automatically.

Although the example shown includes only one file definition, a definition/maintenance run can include many definitions in a single run. The source statements for each definition (file, working storage, table, transaction,❹ or array) are placed back-to-back in the input stream. The only requirements regarding sequence are as follows:

- Table definitions must precede any file or array definitions that reference them using automatic table lookup fields.
- File definitions must precede any transaction definitions that reference the file definition.

If the prerequisite table and file definitions have already been cataloged in the common library, they need not be included in the input for a definition/maintenance run that refers to them. The definitions will be retrieved from the common library as required to perform the necessary cross-validation.

The last section of the listing is the optional glossary for the file definition being cataloged in this example.

```
-----------------------------------------------------------------------------------------------------------------------------------
  RUN    STMT    FILE    O N S D U T R A B S D S L RB R S M R S                START            END              SEQUENCE   LIST
  NAME   TYPE    NAME    L E E I P R E U U R L C S PL J O O F S                SEARCH           SEARCH           73 TO 80    SEQ
                         D W Q R D N P D F T M N T TK T P P O R
-----------------------------------------------------------------------------------------------------------------------------------
(SAMPLE ) (RC)                                                                                                                  1

** CLSAA00  TYPE 0  DEFINITION MASTER   IS BEING PROCESSED.                                                                  *****
-----------------------------------------------------------------------------------------------------------------------------------
   FILE    STMT    FILE    DELETE GLOSSARY RECORD RECORD RECORDS BUFFER MAPPED      MAPPING         MAPPED          SEQUENCE   LIST
   NAME    TYPE    IDENT    FILE   LEVEL   FORMAT  SIZE  PER BLK  SIZE   FILE    INIT      TERM     RECSIZ          73 TO 80    SEQ
-----------------------------------------------------------------------------------------------------------------------------------
(MASTER  ) (FD)             (R)             (F)    ( 80)                                                                        2
-----------------------------------------------------------------------------------------------------------------------------------
   FILE    STMT  SEGMENT  DEL SGMT VRT       SUP SEG  KEY                        CREATOR         EXTERNAL          SEQUENCE   LIST
   NAME    TYPE   NAME    ETE NMBR SEG       DPL KEY CHAR                          ID           TABLE NAME         73 TO 80    SEQ
-----------------------------------------------------------------------------------------------------------------------------------
(MASTER  ) (LS)  (ROOT   )    ( 1)                                                                                              3
-----------------------------------------------------------------------------------------------------------------------------------
   FILE     STMT   FIELD   DEL SGMT LEV FIELD  FIELD FLD SEG FLD DEC CNT  OCCRS OUTPT EDIT   DATA BASE NAME/        SEQUENCE   LIST
   NAME     TYPE   NAME    ETE NMBR NUM LOCATN LNGTH TYP KEY RND PLC SGMT NTIME EDIT  LNGH   COLUMN HEADING         73 TO 80    SEQ
-----------------------------------------------------------------------------------------------------------------------------------
(MASTER  )  (L0) (NEWKEY1 )    ( 1) (1) (   1) (  4) (F) (1)                                                                    4
(MASTER  )  (L0) (NEWFLDA )    ( 1) (1) (   5) ( 16) (C)                                                                        5
(MASTER  )  (L0) (NEWFLDB )    ( 1) (1) (  21) ( 32) (C)                                                                        6
(MASTER  )  (L0) (DUEDATE )    ( 1) (1) (  53) (  6) (C)                                                                        7
(MASTER  )  (L0) (PRINCIPL)    ( 1) (1) (  59) (  8) (P)          (2)                                                           8
** CLSDC02   TYPE 0  THE DEFINITION IS VALID AND HAS BEEN CATALOGED.                                                          *****

                                          DETAILED GLOSSARY BY NAME FOR                                  GLOSSARY PAGE      1
                                            FILE DEFINITION - MASTER
```

Figure 3-108 Sample Definition / Maintenance Run Listing (Page 1 of 2)

```
DATE CATALOGED - 01/15/95 16:43:19                                    DATE LAST UPDATED - 01/15/95 16:43:19
CATALOGED BY -                                                        LAST UPDATED BY -
EXPIRATION DATE -                                                     DATE LAST USED -
FILE IDENTIFICATION =                                                 RECORD FORMAT = FIXED UNBLOCKED
NUMBER OF SEGMENTS IN FILE = 1                                        RECORD SIZE = 80
NUMBER OF FIELDS IN FILE = 5                                          BLOCK SIZE = 80

                                      ********************************
                                      *     SEGMENT  1, LEVEL 1      *
                                      *   SEGMENT NAME = ROOT        *
                                      ********************************

SEGMENT OCCURS N TIMES = 1                                  KEY FIELD 1 = NEWKEY1   TYPE = F   LENGTH = 4
SEGMENT SIZE = 66
NUMBER OF FIELDS IN SEGMENT = 5
-------------------------------------------------------------------------------------------------------------------------------
FIELD   FIELD   FIELD    FIELD   FIELD    DEC    CNT FIELD  EDIT CODES     EDIT    OUTPUT   LINE  DATA BASE NAME/
NAME    TYPE   LOCATION  LENGTH  RNDING  PLACES  FOR SGMT   ( ) ( ) ( )   LENGTH   WIDTH     NO   *** COLUMN HEADING ***
-------------------------------------------------------------------------------------------------------------------------------
DUEDATE   C        53       6                                               6       6
NEWFLDA   C         5      16                                              16      16
NEWFLDB   C        21      32                                              32      32
NEWKEY1   F         1       4                                              14      14
PRINCIPL  P        59       8           2                                  21      21
** CLSGB00  TYPE 0  END OF FILE DEFINITION GLOSSARY.                                                                    *****
```

Figure 3-108 Sample Definition / Maintenance Run Listing (Page 2 of 2)

Chapter

# 4 Field Characteristics and Handling

Fields in VISION:Builder can be either character or numeric. The following tables list the field types and their equivalents in other sources.

**Note:** Z field type does not apply in IMS. Non-key fields can have a type Z format as long as it corresponds to the internal format.

**VISION:Builder**

| Description | Code | Max. Length | COBOL Equivalent | PL/I Equivalent | FORTRAN Equivalent | IMS Equivalent |
|---|---|---|---|---|---|---|
| Zoned | Z | 15 | DISPLAY | Picture | N/A | C |
| Packed | P | 15 | COMP-3 | Fixed Decimal | N/A | P |
| Fixed | F | 4 | COMP | Fixed Binary | Binary | X, F, H |
| Floating Point | E | 4 | COMP-1 | Floating Decimal | Floating | X |
| Time | L,S,P | 8 | COMP-3 | Fixed Decimal | N/A | P |
| | U | 15 | N/A | Fixed Decimal | N/A | P |
| Character | C | 255 | DISPLAY | Character | Character | C |
| Date | D | 4 | COMP | Fixed Binary | Binary | F |

Note that the following table is the same table as on page 3-52 in Chapter 3, *Defining Data*.

**Note:** In this table, a value of blanks in the Date/Time field is not valid in DB2.

**Note:** The VISION:Builder length of a GRAPHIC and LONG GRAPHIC string is (2*n), where n is the DB2 length of the string.

| DB2 Type, Length | | VISION:Builder Type, Length | |
|---|---|---|---|
| CHAR | 1-254 | C | 1-255 |
| VARCHAR | 1-254 | V | 1-254 |
| To update DB2 tables in-place while using VARCHAR fields, use non-MOSAIC processing, which only allows update-in-place processing on a single table. Therefore, if you need to hierarchically join two or more tables, then you cannot use VARCHAR fields. | | | |
| LONG VARCHAR | 255 and larger | V | 255-99H |
| DECIMAL | | P | 1-15 |
| SMALLINT | | F | 1-2 |
| INTEGER | | F | 3-4 |
| FLOAT | | E | 4 |
| DATE | | C or D | 10 or 4 |
| TIME | | C | 8 |
| TIMESTAMP | | C | 16-26 |
| GRAPHIC | 1-127 | C | 2-254 |
| LONG GRAPHIC | 128 and larger | V | 256-99H |

## Character String Fields - Type C

Character string fields are fixed length fields consisting of any number of EBCDIC characters; the maximum length is 255 bytes. A character field is examined and converted to a numeric field when:

**Note:** For these conversions, if a valid numeric data is not found, a value of zero is used.

- An arithmetic operation is performed.
- It is compared to a numeric field type.
- Its contents are replaced into a numeric field type.

See Validation and Automatic Conversion on page 4-10 for details on VISION:Builder automatic conversions.

## Zoned Decimal Number Fields - Type Z

Zoned decimal number fields are checked automatically for valid numeric digits and proper sign before any operation is performed. Refer to IBM's Principles of Operation Manual for a definition of zoned decimal numbers.

All numeric digits (0-9) are valid in zoned fields. Leading blanks are permitted; embedded blanks are invalid. Fields must not exceed 15 digits. A valid sign code must appear in the leftmost four bits of the low order byte of each field. If a zoned decimal field is all blanks, it is treated as having a value of zero and is valid.

## Fixed Point Number Fields - Type F

Fixed point fields are validated during processing according to standard binary conventions. The maximum fixed point binary field size allowed is 4 bytes; the range of values in 4 bytes is -2,147,483,648 to +2,147,483,647. Refer to IBM's Principles of Operation Manual for a definition of fixed point numbers.

## Floating Point Number Fields - Type E

A floating point number field represents a scientific notation, where the decimal number is expressed by an exponent or by a certain power that indicates the actual displacement of the decimal point. For example, a numeric field with 123000000.0 can be expressed in scientific notation as $1.23 \times 10^8$ or 123E6. In VISION:Builder, floating point fields are 4 bytes in length. Refer to IBM's Principles of Operation Manual for a definition of floating point numbers.

## Time Fields - Type L, S, or U

Time fields are numeric fields consisting of data belonging to a three-unit system of measurement, such as hours-minutes-seconds, yards-feet-inches, and pounds-ounces-drams. These fields are described in detail in Time Fields (Types L, S, or P) on page 4-8.

## Date Fields - Type D

Date fields are integer fields containing a date known as a Lilian Date. Lilian Dates are integers representing the number of days since October 14, 1582. Conversion of character representations of dates to Lilian and vice versa are automatically performed using the callable services of Language Environment from IBM LE/370.

## Packed Decimal Number Fields - Type P

Packed decimal number fields are checked automatically for valid digits and proper sign in the rightmost four bits of the low order byte before any operation is executed. The maximum size allowed for a packed decimal field is 29 digits or 15 bytes. Refer to IBM's Principles of Operation Manual for a definition of packed decimal numbers.

# Extended Packed Decimal Fields

Extended packed decimal fields are packed decimal fields with a length greater than eight. They can be defined with a length of up to 15 bytes (up to 29 digits). VISION:Builder performs all the automatic conversions and editing for extended packed fields, as it does for any other field type. If no override edit pattern is provided by the user, VISION:Builder prints the value of the extended packed field (up to 29 digits, plus all necessary punctuation) using a default edit pattern.

A decimal constant can contain a maximum of 15 digits. This means that an extended packed value cannot be placed into an extended packed field using a decimal constant as an operand.

An implicit temporary field (see Implicitly Defined Temporary Fields on page 4-17) inherits default attributes. It is possible for an implicit temporary field to be set up with the attributes of an extended packed field. It is important to know these attributes when you are using this temporary field in arithmetic operations, especially when multiplying and dividing with extended packed numbers.

When using an extended packed decimal field in an arithmetic multiply or divide, special consideration must be given to the operands involved.

- Operand A must be the extended packed field. It is automatically converted to a packed field with a length of 16 and a scale large enough for the fields involved in the operation. The number of leading zeros in the converted value of Operand A must be greater than or equal to the length of Operand B. Note that it is the length of the field used as Operand B, not the length of the value in Operand B.
- Operand B must not be an extended packed field (if it is, a MK4DR01 type 5 message will be issued). Operand B cannot be an implicitly defined temporary field because it would inherit the attributes of Operand A and become an extended packed number resulting in the MK4DR01 error.
- Operand C must be an extended packed field to hold the large result.

To achieve the maximum precision when multiplying (or dividing) an extended packed number in Operand A, ensure that all three operands are packed fields and define Operand B with the smallest length possible. This avoids the conversion to a common type, length, and scale. The result is that Operand B will retain the length it was defined with for the operation, requiring fewer leading zeros in Operand A.

# Variable Length Fields (Type V)

A variable length field is a variation of a character string field that allows an increase or decrease in the length of a field according to the amount of data it contains. The length of this type of field is adjusted to eliminate trailing blanks. Data is not lost due to truncation and storage space is conserved. These fields can be up to 9,900 characters long, not including the 4-byte header.

A variable length field consists of a 4-byte header followed by character data. The header is composed of a 2-byte current length indicator with two additional bytes reserved for VISION:Builder internal use. The length you specify for the field does not include the 4 bytes needed for the header. When a variable length field contains no data and its current length indicator shows the value zero, its maximum length will actually occupy 4 bytes (the header). The header is reserved for use by VISION:Builder and must not be altered (see Figure 4-1).

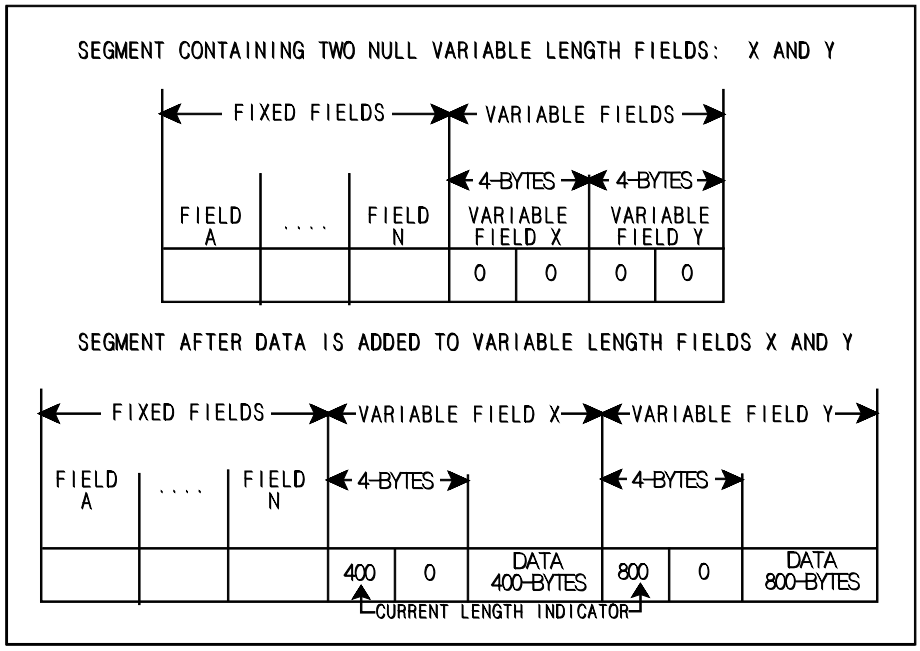



Figure 4-1 Variable Length Fields in Segments

There can be any number of variable length fields in the file. They can be defined in any segment, at any level. However, all such fields are placed at the end of the segment and their location within a segment is computed by VISION:Builder. The location of the first variable length field in a segment is always fixed and follows the last fixed length field in the segment.

If a segment contains more than one variable length field, the positions of those following the first are dependent on the length of the preceding variable length field(s). Overlapping field definitions are not possible with variable length fields.

Even though VISION:Builder processes records created by other languages containing compatible variable length fields, you must remove all trailing blanks from these fields regardless of whether they are accounted for in the field length count.

Variable length fields can exist in the master file, coordinated files, subfiles, and as temporary fields.

When a temporary variable length field is defined explicitly, its maximum storage is allocated and maintained in storage. The location of other fields in storage is not affected by changes in the length of a temporary field. If an initial value is supplied for the field in the FIELD statement, its length determines the value of the current length indicator for the field; therefore, the length will not be null. When defined implicitly, the rules for implicit temporary fields apply. See Temporary Fields on page 4-16 for definitions of implicitly and explicitly defined temporary fields.

You can update variable length fields using transactions and procedures. When the length of a variable length field is altered, its current length indicator (the first 2 bytes of the field) is automatically modified accordingly and the record containing the field is altered to compensate for the change in field length. A variable length field becomes invalid if an operation intended to expand it causes the maximum defined record size to be exceeded. It also becomes invalid if an operation attempts to alter it using an invalid field.

Variable length fields cannot be defined in a transaction definition, but a character field in a transaction definition can be used to update a variable length field. Transaction action codes are provided to create and maintain variable length fields with transactions. There are four actions that can be performed on variable length fields during transaction processing: concatenate (type X), replace (type P), clear to blanks (type B), and conditional replace (type R). See Chapter 10, *Transactions* for more details on these actions.

## Variable Length Fields in Reports

Variable length fields are printed in reports by folding them within their specified column width until the entire field is printed. If a variable length field exceeds its column width, it is split between words (at a blank), if possible. The printing is continued on the next line with the first non-blank character. If a word (a character string without an embedded blank) exceeds the column width, it is split. More than one variable length field can be output in one detail line and the line can be folded as with other types of fields.

The column width is specified in the file definition by using the Length entry of the VISION:Workbench for DOS Output Edit popup window on the Segment Definition window. It can be overridden for a report by changing the Output Edit entry for the field on the Report Detail window in VISION:Workbench for DOS.

Variable length fields can be printed in standard, formatted, or formatted sectional reports, but will only fold in detail lines. Variable length fields cannot be summarized. They cannot be used as control breaks or sort fields. On report output, null fields show as blanks; invalid fields show as the character defined for this in M4PARAMS; the default is asterisks (*).

Normally, a variable length field cannot be split between pages of a report. For example, if ten lines are required to print a variable length field and only six lines remain on a page, the remainder of the page is skipped and the field prints at the top of the next page. Because this can result in inefficient use of space, an override method is provided, allowing fields to be split between pages:

**Note:** The Blank and S entries cause breaks in variable length fields between words and left-justification of the folded data by removing all blanks between words at the break point.

Blank — All occurrences of the field are printed.

A variable length field automatically folds to the next line within the column width. The fold occurs between words. Any leading blanks are removed from the beginning of the next line. Printing of variable length fields begins on the following page if the entire field will not fit on the remaining lines of the current page.

S — Same as the blank entry, except that a variable length field begins printing on the current page and is split between report pages if the bottom of the current page is reached before the entire variable length field is printed.

**Note:** The H and B entries cause breaks in the variable length field at units equal to the column width. No justification occurs because no blanks are removed.

H — A variable length field folds into multiple report lines in units equal to the column width without regard to word boundaries. Blanks are not removed; all blanks are preserved. Printing of variable length fields begins on the following page if the entire field will not fit on the remaining lines of the current page.

B — Same as the S entry, except that a variable length field begins printing on the current page and is split between report pages if the bottom of the current page is reached before the entire variable length field is printed.

### Restrictions on Using Variable Length Fields

- They can interact only with other variable length fields and with character fields. They cannot be used in an arithmetic operation even if all of the fields involved are character or variable length fields.
- They cannot be used in file definitions used as M4WORKnn or M4LINKnn.
- They cannot be defined in a transaction definition. They are maintained in a master file using character transaction data fields.
- They cannot be used as an argument for a table lookup operation. However, they can be the result of a basic table lookup operation if the table result type is a character string.
- When output to subfiles, they must follow fixed length fields.
- They cannot be summarized.
- They cannot be used as control breaks or sort fields in reports.
- They cannot exist in file definitions used with files processed using MOSAIC processing. They can be updated using non-MOSAIC processing, but then you can only use a single table (you cannot hierarchically join two or more tables with non-MOSAIC updating).

## Time Fields (Types L, S, or P)

Time fields are numeric fields consisting of data belonging to a three-unit system of measurement, such as hours-minutes-seconds, yards-feet-inches, and pounds-ounces-drams. The choice of the measurement system is an M4PARAMS parameter at installation time. Refer to the *VISION:Builder for OS/390 Installation Guide* (OS/390 users) or the *VISION:Builder Installation Manual for VSE* (VSE users).

The following table illustrates the field types, their formats, and their minimum and maximum lengths when the unit system is time.

**Note:** A packed field can be used to represent time data belonging to a two-unit system of measurement.

| Field Type | Format (interpreted) | Field Length (in bytes) Minimum | Maximum |
|---|---|---|---|
| L (HR-MIN-SEC) | HHH...HHMM SS NNN...N (MM and SS must be two digits) | $\left[\frac{Nn + 7}{2}\right]$ * | 8 |
| S (MIN-SEC) | MMM...MMSS NNN...N (SS must be two digits) | $\left[\frac{Nn + 5}{2}\right]$ * | 8 |
| P (Packed Decimal) | SSS...SSNNN...N | 1 | 15 |

Nn = Number of decimal of seconds digits (Nn ≤ 9).
* Use only the integer portion of the number.

The colon, or other installation parameter, must be used to separate the hours from the minutes and the minutes from the seconds.

VISION:Builder outputs a type L time field in the following format (colon is installation option):

```
(sign) hh. . .h:mm:ss.nnnn
```

VISION:Builder outputs a type S time field in the following format (colon is an installation option):

```
(sign) mm. . .mm:ss.nnn. . .n
```

All time data is maintained in packed decimal representation in VISION:Builder.

In an ASL statement, a time literal constant is enclosed in single quotation marks and preceded by the letter T. On a FIELD statement, the initial value for a time field (type L or type S) is coded in the same notation.

## Time Data Conversions and Arithmetic Operations

Time data can be treated as any other data type in VISION:Builder. VISION:Builder can convert time data to any other numeric type using the rules outlined in Validation and Automatic Conversion on page 4-10. Time data is converted to a packed form before affecting the conversion. When time data is converted to character type, the resulting character string contains the data in time format complete with a minus sign, colons (if time data was input as a literal constant), and a decimal point. Leading zeros are suppressed up to the decimal point.

All arithmetic operations and logical operations on time data are performed by converting the data to seconds and decimals-of-seconds and converting the result to a field type specified as the result field.

All non-time data is assumed to contain time information in the form of seconds and fractions of a second when used with time data fields or time literal constants. Therefore, attempts to combine time data with other types can produce unwanted results, particularly if character type fields are involved.

## Date Fields (Type D)

Date fields are numeric fields containing a date value known as a Lilian Date. A Lilian Date represents the number of days since 14 October 1582. The date 15 October 1582 is represented as a value of 1 and each day thereafter increments the value by 1. A Date field is stored as a 4-byte integer and it can represent dates until 31 December 9999. The difference in days between two dates can be derived by subtracting a Date field from another Date field. To derive a date some number of days in the future to or prior to a given date, add or subtract the appropriate number of days from the Date field as the case can be. Since date fields only occupy 4 bytes of storage, conversion from other date formats usually does not require the expansion of the underlying file or database records.

Conversion of character representations of dates to Lilian and vice versa are automatically performed using the callable services of Language Environment from IBM. Selected LE services are invoked automatically by VISION:Builder whenever any conversion between a Date field and its character representation is performed. These automatic conversions assume that the character representation of a date conforms to the installation parameters specifying the order of the year, month, and day within the date. In addition, if the character representation is 8 or more characters long, the date is assumed to contain a 4-digit year. Otherwise, a 2-digit year will be assumed. This is true for both the case when a Date field is being converted to its character representation or when a character string is being converted to a Date field. LE services can be invoked explicitly to perform specialized date related conversions and processing of Date fields within an application.

Edit patterns used with Date fields are assumed to be date format pictures, for example MM/DD/YY. Refer to the IBM Language Environment Reference Manual for a list of valid picture characters.

## Validation and Automatic Conversion

VISION:Builder performs automatic actions on numeric, alphanumeric, and alphabetic field types. Each field is checked for validity prior to any operation, according to automatic internal validation criteria.

In application development, using different field types can critically affect operation performance. To maximize processing efficiency, logical operations should use fields of the same type. Arithmetic operations should use packed

decimal fields or fixed point binary fields of equal scale. In other cases, VISION:Builder will internally convert the fields to a compatible type according to the rules specified in the tables on pages 4-11 and 4-12.

## Rules for Conversion in Arithmetic and Comparison Operations

The following table summarizes the field conversions for comparison and arithmetic operations.

| Field Type 1 | Field Type 2 | Both Fields May Be Changed To | In Accordance With Rule |
|---|---|---|---|
| C | C | P | A |
| C | D | D | X |
| D | C | D | X |
| Z, P, F | C | P | B |
| Z, P, F,D | Z or P | P | C |
| Z, P | F or D | P | C |
| F or D | F or D | Binary or P | D |
| C, Z, P, F, D | E | E | E |
| D | C | D | X |

Key: C=Character, E=Floating Point, F=Fixed, P=Packed, Z=Zoned Decimal, D=Date

Prior to the execution of all logical, arithmetic, and replace operations, VISION:Builder checks fields for valid data, aligns fields, and makes conversions according to the table starting on page 4-11 and the rules that are described in the table starting on page 4-12.

**Note:** In transaction processing,❹ if a master record is created or a segment inserted, all fields except the record or segment key fields are initialized to blanks for character string data and zeros for numeric data. When fields overlap, they are initialized based on the data type of the last overlapping field defined.

When the result field of an arithmetic or replace operation does not have enough decimal places to hold the entire result and rounding is not specified, VISION:Builder truncates those digits to the right of the decimal point. If the result field is not large enough to hold all significant digits to the left of the decimal point, the field is flagged as invalid. See Invalid, Missing, and Overflow Fields on page 4-18.

A In conversion for comparison or arithmetic operations on two-character fields of different lengths, the shorter field is padded with trailing blanks before performing the operation. In comparisons, no conversion takes place. In arithmetic operations, both fields are converted to numeric values in packed decimal format.

B In conversion for comparison or arithmetic operations on a character field and a numeric field, the character field is scanned from left to right until the first numeric character or sign is encountered. The scan continues until the first non-numeric character (other than properly placed decimal points, commas, or exponent signs) is encountered or until the limit of 15 digits or the end of the field is reached. Decimal points are aligned, the fields are padded with zeros, and field type conversions to packed decimal are performed, as necessary. If no valid numeric data is found, a value of zero is used.

C If numeric fields in an operation consist of packed decimal, fixed point, and/or zoned decimal or if all fields are zoned decimal, they are changed to packed decimal format before the operation is executed. In conversions for comparison or arithmetic operations on numeric fields, the fields are aligned on the decimal point. To accomplish this, leading and trailing zeros are added as necessary and significance is not lost.

D Arithmetic operations on two fixed point binary fields are performed in binary format. If the two are of different scales, they are converted to packed.

E If one field is not floating point numeric and the other field is floating point, the non-floating point field is changed to floating point before performing the operation.

X Operations involving date fields are performed in binary format. Character fields are converted to date fields before performing the operation.

## Rules for Conversion in Replacement Operations

The following table summarizes the field conversions for replacement operations.

| Sending Field Type | Receiving Field Type | Replace Action Rule |
|---|---|---|
| Z, F | P | F |
| P, Z, F | C | G |
| C | C | H |
| E | C | I |
| Z | Z | J |
| Key: C=Character, E=Floating Point, F=Fixed, P=Packed, Z=Zoned Decimal | | |

| Sending Field Type | Receiving Field Type | Replace Action Rule |
|---|---|---|
| C | Z | K |
| P, F, E | Z | L |
| E | Z, P, F | M |
| C | D | Y |
| D | C | Z |
| Key: C=Character, E=Floating Point, F=Fixed, P=Packed, Z=Zoned Decimal | | |

Prior to the execution of all logical, arithmetic, and replace operations, VISION:Builder checks fields for valid data, aligns fields, and makes conversions according to the table starting on page 4-12 and the rules that follow.

F When a zoned decimal or fixed point binary field is replaced into a packed decimal field, the generated sign is C (positive) or D (negative).

G When data is converted from a numeric field (zoned, packed, or fixed) to a character field, leading zeros are suppressed from the character receiving field (except if the conversion results from an action code in a transaction definition). If there are implicit decimal places in the numeric field, an explicit decimal point appears in the same relative position in the receiving field. A negative sign, if applicable, displays to the left of the first non-blank character. The character receiving field must be large enough to accommodate the applicable decimal point and/or negative sign (no commas). A zero value result contains a single right-justified zero.

H When a character field is filled or replaced from another character field, left-justification is performed. If the sending field is longer, the rightmost characters of the sending field are truncated; if the sending field is shorter, the rightmost portion of the receiving field is padded with blanks.

I Floating point fields are converted to the exponent (scientific notation) format when converted to character fields.

J When a zoned decimal field is replaced into another zoned decimal field, any leading blanks in the result field are changed to zeros (for example, 40F0 is changed to F0C0). Resulting signs are forced to C (positive) or D (negative).

K When a character field is replaced into a zoned decimal field, the result field has a sign of either F (positive) or D (negative).

L When a packed, fixed, or floating number field is replaced into a zoned decimal field, the generated sign in the result field is C (positive) or D (negative).

M When converting floating point fields to another numeric field type (zoned decimal, packed decimal, or fixed point binary), the result field is converted to fixed point binary; the fixed point field is converted to the wanted field type. Overflow occurs if the converted floating point field exceeds the range of -2,147,483,648 to +2,147,483,647.

Y When converting character fields to date fields, the character field is assumed to contain a date whose year, month, and day order match the date format specified in M4PARAMS. Furthermore, it is assumed that the character field does not contain any separators. If the character field is eight characters or more in length, it is assumed that the date contains a 4-digit year. Otherwise, it is assumed that the date contains a 2-digit year in which the LE service CEESCEN can be used to set the appropriate century.

Z When converting date fields to character fields, the character field will contain a date whose year, month, and day order will match the date format specified in M4PARAMS. Furthermore, the character field will contain the separator specified in M4PARAMS to separate the year, month, and day subfields. If the character field is 10 characters or more in length, the date will include a 4-digit year. Otherwise, the character field will only contain a date with a 2-digit year.

# Output Edit Length

Output edit length specifies the number of characters necessary for printing a particular field. VISION:Builder automatically computes an edit length large enough for the maximum number of positions required if this specification is left blank. However, if the number is specified by you, it must be large enough to allow for any additionally requested edit characters. The output width that appears on the glossaries specifies the width that VISION:Builder allows on an output report. The output width is the maximum of the output edit length and the column heading length. If a field is too large to print in the output width provided, VISION:Builder prints a plus sign (+) on the report for that field. The plus sign is an installation option.

The following information illustrates and summarizes procedures for the various types of fields.

## Variable Length Field

The output edit length of variable length fields can be entered using a file definition, temporary field definition, or report definition. If it is not specified in any of these ways, the output edit length defaults to the maximum length that the current report allows. The following factors affect the default output edit length: page width, existence of summary labels and/or numbers, spaces before columns, and the width of other fields on the print line.

## Character Field

In the VISION:Workbench for DOS Segment Definition window, in the Output Edit, Length entry, enter a number from 1 to 99 to specify the number of characters for the width of the data column in a report. If the maximum length exceeds 99, do not make an entry. Decimal points, commas, and output edit codes are not used.

Character string fields, when printed, can be truncated from the right or padded with trailing blanks, as appropriate. No other changes are made to the contents. This is true of all data that prints from character fields.

A zero numeric value prints as a right-justified zero (after a replace from a numeric field to a character field) if there are no decimal places in the source field. If there are decimal places in the source field, it prints as a decimal point followed by zeros. A negative sign, if applicable, appears to the left of the first non-blank character.

VISION:Builder computes a default output edit length for character fields equal to the assigned field length in the file definition or the width of the column heading text, whichever is greater. For example, assume a character field where the field length is specified as 130 bytes and the column heading is 20 bytes long. For character fields, 1 character equals 1 byte; therefore, the computed default output edit length for this field is 130 positions.

## Packed, Zoned, and Fixed Point Fields

In the VISION:Workbench for DOS Segment Definition window, in the Output Edit, Length entry, enter a number from 1 to 30, equal to the maximum number of positions necessary to print this field. The number entered is the sum of the following items:

- The maximum value (number of digits) that can exist in the field. If the decimal places are specified for the field, this count must be equal to or greater than the number of decimal places specified.
- The decimal point, if decimal places are specified.

**Note:** If the edit suppress code (Z) is specified in the leading edit code column, do not allow positions for the printing of the decimal point or commas.

- Commas to the left of the decimal point or to the left of the low order digit when decimal places are not specified. The number of commas required to print a field is the integer value of the expression:

$$\frac{(\text{number of digits}) - (\text{number of decimal places}) - 1}{3}$$

- Specified edit codes are detailed in the VISION:Workbench for DOS online HELP.

If the output edit length is not specified or the entry is greater than 30, VISION:Builder computes a default output edit length as the sum of:

- The specified field length (converted to digit positions according to field type). For example, a fixed point (type F) field with specified field length of 4 bytes when converted to digit positions would equal ten.
- Decimal point, commas, and special edit codes as specified under specifications above.
- The larger of the column heading text length or this sum is used as a default output edit length.

### Floating Point

In the Segment Definition window, in the Output Edit, Length entry, enter a number between 7 and 30 to include the two signs, decimal point, E, and two exponent digits. Floating point number fields always print in standard notation, where Xs represent the fraction and Ys the exponent:

± .XXXXXX E ± YY

If the length exceeds 13, leading blanks print.

Floating point summaries are not truncated to integer values; instead, they print with no loss of significance (other than normal rounding errors inherent in floating point conversion).

If output edit length is not specified or the entry is greater than 30, VISION:Builder assigns a default output edit length for type E fields equal to 13.

### Date Field

The default edit length is ten and the minimum edit length that can be specified is eight.

## Temporary Fields

Temporary fields are useful as result fields, when performing computations and as communication between multiple procedures in a run. One procedure can define a temporary field and subsequent procedures can use the value of the temporary field that was set in the first procedure. The value of a temporary field is initialized prior to processing.

You define the temporary field for the duration of a particular run. Temporary fields can be either implicitly or explicitly defined.

## Implicitly Defined Temporary Fields

Implicit temporary fields are temporary fields whose attributes are set up by default. All you specify is a T qualifier and the field name when you reference the field (for example, T.FLDNAME).

When encountered for the first time, such a field is automatically assigned the attributes of a base field (the leftmost defined field in the LET statement).

If there are no base fields in the LET statement, the temporary field is assigned these default attributes:

- Field length of 8 bytes.
- Field type of P (packed).
- Eight decimal places.
- Output edit length of 19 characters.
- Leading edit character blank.
- Trailing edit character blank.
- Filling edit character blank.
- Field name used as column heading, if reported.

Figure 4-2 shows how a temporary field is implicitly defined using ASL. In this case, the temporary field T.TEMP1 takes on the attributes of the base field CUSTNO field.

```
LET T.TEMP1 = CUSTNO
```

Figure 4-2 Defining a Temporary Field Implicitly

Since you do not maintain control over the attributes when implicitly defining temporary fields, you may want to define them explicitly.

## Explicitly Defined Temporary Fields

Explicit temporary fields are temporary fields explicitly defined using an ASL FIELD statement. Refer to the *VISION:Builder ASL Reference Guide*. This means the attributes of the field are spelled out by you. These definitions are not cataloged in the common library.

Several attributes of the field can be defined, such as length, type, and output edit characteristics. An initial value can also be set.

When referring to a temporary field in any other statement, you must use the temporary field name with its qualifier T.

Figure 4-3 shows how a temporary field is explicitly defined using ASL. Note that an initial value is set and column headings are assigned.

```
TEMP2: FIELD TYPE C LENGTH 10 INIT '100' ,
       HEAD 'CUSTOMER' 'NUMBER'
```

Figure 4-3 Defining a Temporary Field Explicitly

# Invalid, Missing, and Overflow Fields

## Invalid Fields

During the course of arithmetic or logical operations or conversions, a field can become or be interpreted as invalid for a variety of reasons:

- Illegal field data. Non-numeric data where numeric data was expected. Replacing a missing field (a field that does not exist in the record) into a result field.
- Division by zero.
- Field overflow. Result fields not large enough to accommodate resulting data, causing result fields to become invalid.
- Invalid or missing sign.

When VISION:Builder encounters an invalid field, it is flagged as invalid but its original contents are not changed. This invalid property is propagated and other result fields that depend upon the invalid field become flagged as invalid. Again, field contents are not destroyed.

When there is an attempt to print an invalid field, an asterisk (or the installation-specified value for invalid fields) prints. An asterisk also prints if a field is defined as a numeric field type in a file definition, but the data in the field does not match the format.

A field that becomes invalid during a control procedure remains invalid for the duration of that procedure only. Any work area field that becomes invalid in the first procedure is revalidated prior to executing subsequent procedures with no indication that it was ever invalid. Revalidation of a field does not affect field contents; it merely resets the field property indicator to valid.

The exception to this rule is that of temporary fields and flags. Since temporary fields and flags are designed to communicate between procedures, they carry their invalid property between procedures.

All invalid fields can be revalidated by storing valid data into the field. Revalidation does not occur if the result field is partial fielded during the store, even if the partial field defines the entire field.

A work area field that is invalid before the first time it is referenced (for example, a record from the master file that contains alphabetic characters in a field defined as packed decimal) remains invalid as long as that record is in memory, unless it is revalidated by storing valid data in the field.

If a field in a lower level segment occurrence becomes invalid, it remains invalid as long as it is currently being processed. If it is reprocessed through a subsequent loop or FIND function (FS operation) in the same procedure, it is revalidated at that time.

An invalid field output to a subfile will contain the maximum positive value that the field can contain. In the case of character strings, this value is a string of X'FF' bytes.

Subroutine procedures inherit fields that are in segment occurrences currently being processed by the procedure or subroutine procedure that called it. If a subroutine procedure inherits an invalid field, the field can be revalidated and returned as valid to the procedure that called it.

Similarly, an inherited field can become invalid in a subroutine procedure and be returned as invalid to the procedure that called it. If a non-inherited field becomes invalid in a subroutine procedure, it will stay invalid only for the duration of that subroutine procedure.

To determine whether a field is invalid, compare the field to itself in a procedure statement. If invalid, the equal compare fails. If valid, the equal compare is true.

## Missing Fields (Non-Existent Fields)

Fields in VISION:Builder are said to be missing or non-existent when either segment occurrences for the field do not exist in the record or its value is not available at a particular time during procedure processing.

For example, if there are no occurrences of a defined segment type subordinate to the parent segment currently in memory, VISION:Builder treats the child segment as missing. A dash (-) is printed for each field from the child segment on the report. Also, in sequential coordination, during a miss condition, coordinated file records can be unavailable and fields referred to in the coordinated file are flagged as missing. Missing fields print with a dash on reports. The dash is an installation option.

When missing fields are output to a subfile, VISION:Builder outputs a zero value for numeric fields and blanks for character string fields.

You can determine if a field is missing by testing if the field is equal to itself in a procedure statement. If missing, the test fails; if not missing, the test is true.

Replacing a missing field into result fields causes the result field to become invalid.

## Field Overflow

VISION:Builder prints a plus sign (+) for fields that are too large to fit in the output column of a report. Additionally, a plus sign can be printed in some cases for summaries computed in the report step by VISION:Builder that are invalid. The plus sign is an installation option.

# Partial Fielding a Character Field

In a VISION:Builder application, you can access a subset or portion of a character or variable length field and restrict processing to just that portion. In ASL syntax, the built-in partial field (PF) function is used to isolate part of a character or variable length field for use in any replacement or logical expression. You supply the field name, start location, and length.

For example, the following syntax allows you to access the string CDEFGH in a field named ALPHABET, which is 26 characters long and contains the letters A through Z. Refer to the *VISION:Builder ASL Reference Guide* for details on the PF function:

```
PF(ALPHABET START 3 LENGTH 6)
```

Chapter

# 5 The Application Cycle

Chapter 2, *Application Development System* introduced VISION:Builder as a high level language that you can direct with simple statements to create applications. This simplicity is based upon the fact that VISION:Builder assumes a basic application cycle derived from the specific inputs and outputs defined by your application. The statements that specify the basic inputs and outputs for an application are referred to as Run Control statements. With only a few simple Run Control statements, you can create an application that copies a file.

The definitions that describe the data that the application automatically reads and writes and makes available to the application in work areas are the basic component of the application. The automatic flow of records through these work areas is depicted in a simple flow diagram and described later in this chapter. These definitions are cataloged in the common library and can be shared by many applications. The details of these definitions are described in Chapter 3, *Defining Data*.

A VISION:Builder application can be customized by introducing additional components, known as procedures, reports, and subfiles, into your application. These components interact with the basic application cycle to provide traditional data processing functions for performing data selection and manipulation, report formatting, and secondary file output. This chapter clarifies the relationship of these components to the automatic processing cycle and the flow of records through the work areas.

The components of an application run are shown in Figure 5-1. The files in the figure are restricted to a master file, a transaction file,❹ a report file, and subfiles so that you can more easily understand the application cycle and record flow. Later chapters will expand upon this introductory information to show how an application can be constructed to handle significantly more complex processing requirements.

**Note:** In Figure 5-1, old and new master files are the same physical file if updating-in-place.

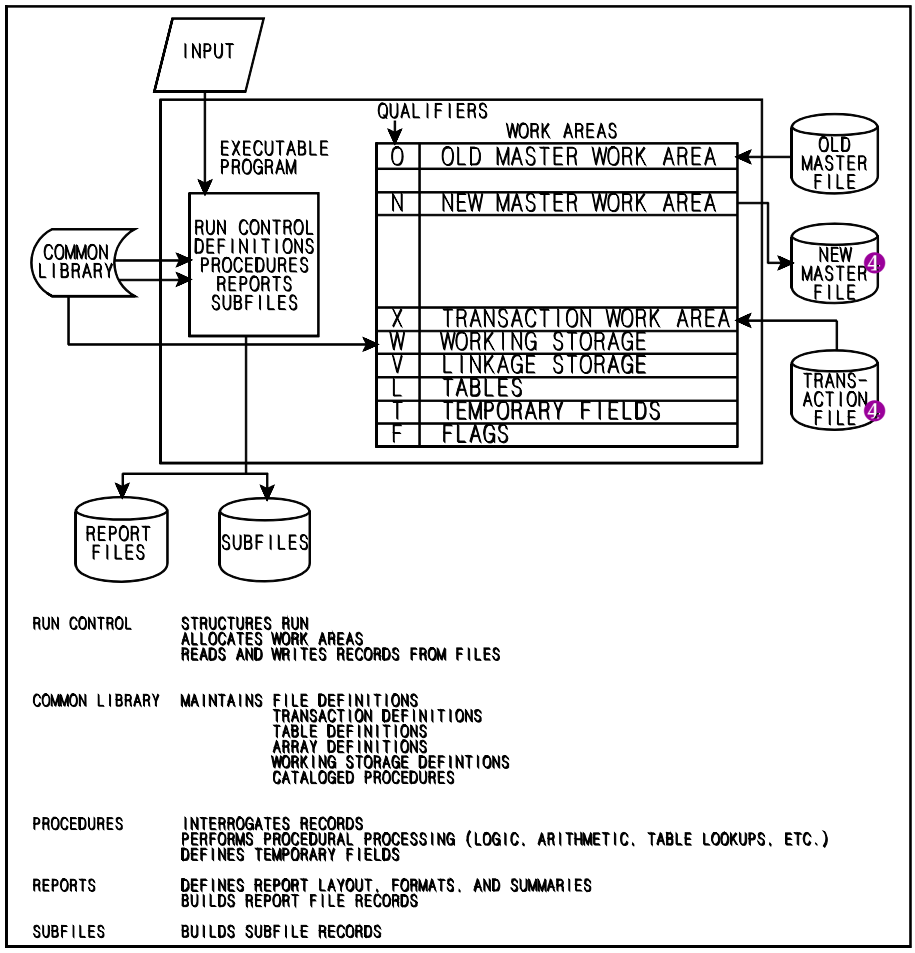



Figure 5-1 Basic Processing Components

# The Common Library in Basic Processing

VISION:Builder processes files and searches tables according to physical and logical data structures cataloged in a common library. As a result, you must catalog file and table definitions in a common library before processing files or searching tables in an application run. This cataloging is performed in a definition maintenance run and is described in detail in Chapter 19, *Cataloging*.

In an application run, VISION:Builder automatically retrieves the required definitions and tables as referenced by name in the run control and procedure components of the application. All of these components are automatically brought together at run time to create an executable application program. This approach keeps data definitions independent of programs and results in significant advantages.

A physical change to a file (for example, adding a new field and changing a record length) does not require that each program be modified and recompiled as with traditional procedural languages. Only the central definition in the common library need be changed; it will be automatically accessed and included the next time any program using that definition is executed. This independence results in significant savings of both programmer and machine time.

# Application Cycle Overview

This section provides an overview of the basic processing cycle built into VISION:Builder. The information presented assumes only the basic elements of the cycle. Subsequent chapters on input files, output files, and transaction processing❹ illustrate how this basic cycle is extended to cover the needs of applications that require more complex processing cycles.

Within the basic cycle, the master file is the primary file in an application run. The term master signifies only that the file so designated is the primary file in the application run. There is no significance to the term with respect to data or information attributes of the file. The master file must be specified as the old master file and can be updated-in-place❹ or output as a new master file❹ along with an optional audit file❹ of deleted master file records.

You can use the ASL Run Control command group to specify the common library file definition to be used to process the master file records and other master file processing parameters, as shown in Figure 5-2. You can also use it to specify additional application parameters. Together, these entries constitute the component of an application that, in very general terms, defines the inputs and outputs to be included into the automatic processing cycle of a VISION:Builder application.

```
CONTROL  SORT EXTERNAL  SUMMARIZE   SAMPLE
FILE MASTER INPUT  CUSTMAST
FILE MASTER OUTPUT
FILE AUDIT
FILE REPORT
```

Figure 5-2 Sample Run Control Command Group

With the specifications provided, the master file is processed sequentially, retained records are written to the new master file,❹ deleted records are written to the audit file,❹ and selected and formatted records are written to the report file. The report file records are described later in this chapter. For more information about processing options, see Chapter 10, *Transactions*.

The master file, in this example, is processed according to the flow diagram in Figure 5-3. VISION:Builder first reads an old master record into the old work area and immediately copies that record to the new work area. Both work areas are available for processing. As the work areas are processed, fields can be referenced and formatted for report file records.

After processing the work areas, the new work area is written to the new master file if it is retained or to the audit file if it is deleted.❹ Another old master record is read into the old work area and copied to the new work area and processing continues.

In this manner, VISION:Builder processes the master file until the old master file reaches end of file — one record at a time and in one direction only. Notice that all of the processing in Figure 5-3 (except procedures and reports) is accomplished by defining the specific input and output files for the application (see Figure 5-2). This is a good example of VISION:Builder's automatic capabilities. The entire structure and flow of the application run is specified by defining the input and output files.

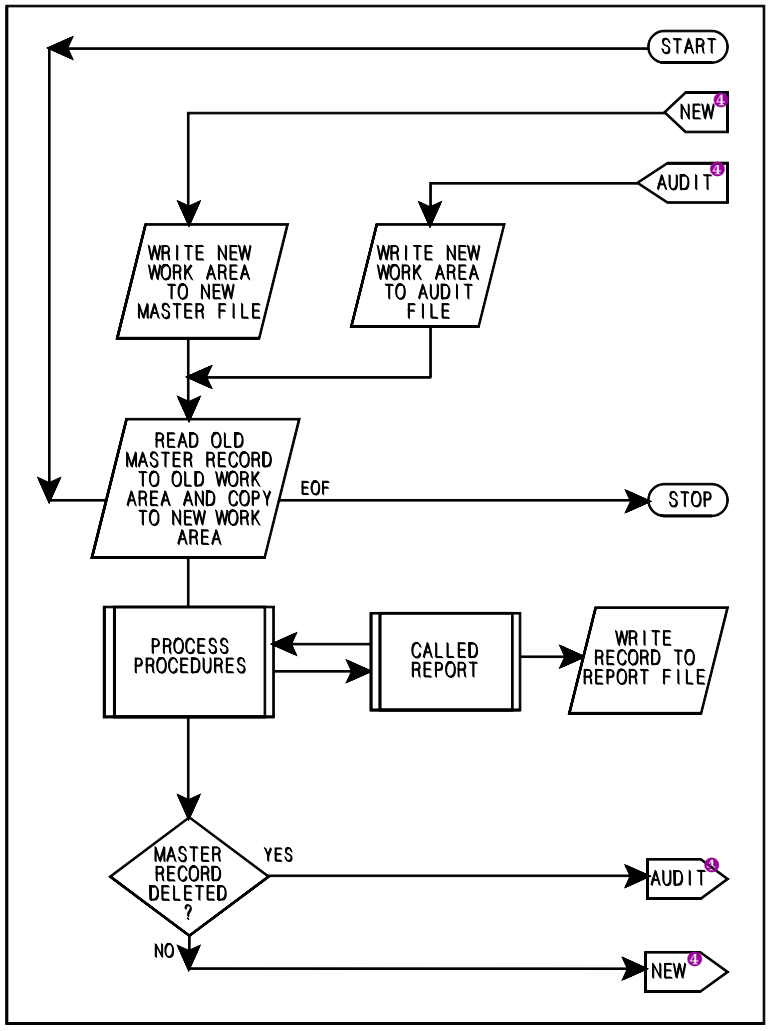



Figure 5-3 Basic Application Cycle

Finally, the application run is divided into process, sort, and report steps to complete the basic application processing. These steps can be run as individual job steps or their function can be combined into a single step to optimize computer resources. The concluding sections of this chapter discuss these considerations.

# Basic Processing Work Areas

The work areas are central to the application run. The master file, audit file, working storage records, linkage areas, temporary fields defined by procedures, and VISION:Builder flags are all accessed in work areas. After input records are read into the work areas by VISION:Builder, they are available to your application's procedures that can modify the records or select them for report files or subfiles. The work areas are the common ground between the VISION:Builder automatic file processing cycle and the procedures, reports, and subfiles that comprise your application.

## Work Areas, Procedures, Reports, and Subfiles

A procedure defines a processing algorithm to be performed using the data provided by VISION:Builder in the respective work areas. Reports define the content and format of printed output. Subfiles define a subset of data contained in the respective work areas for output as an external file. Subfiles can in turn be used as input files by other applications.

Statements within a procedure, report, or subfile component reference the work areas by qualifier and name (see Figure 5-4). For example, an O qualifier references the old work area, an N qualifier references the new work area, and a T qualifier references temporary fields. The field name references a particular field within the work area identified by the qualifier.

Procedure statements can select records (contained in the work areas) for the report files or subfiles. The detail specifications of each report and subfile select which fields in the work areas to output. Procedure statements can be used to update work area records or communicate with other procedures using work areas, especially through temporary fields and working storage.

For more complex applications, the working storage and linkage sections can be used to communicate with other application programs and VISION:Builder flags can provide status information on the VISION:Builder process. Working storage is described in Chapter 3, *Defining Data*; linkage sections are described in Chapter 18, *External Program Linkage*; temporary fields are described in Chapter 4, *Field Characteristics and Handling*; and flags are described in Flag Fields on page 5-6.

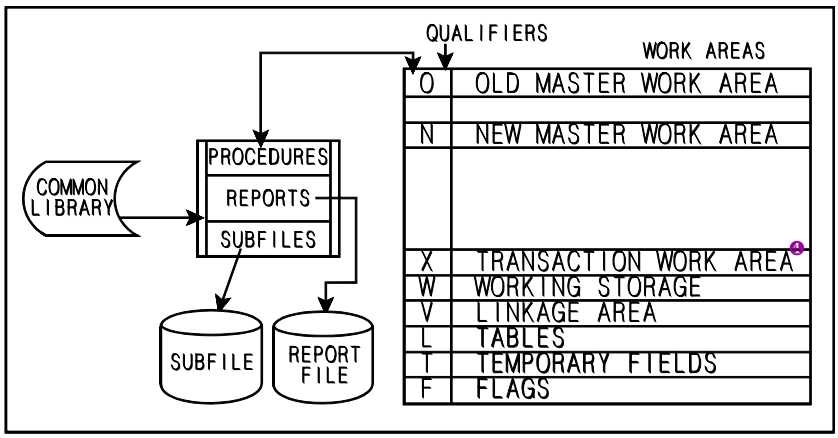



Figure 5-4 Work Areas and Procedures, Reports, and Subfiles

Temporary fields, explicit or implicit, are available to the defining procedure or to subroutine procedures, reports, and subfiles using the T qualifier. See Procedures on page 5-8 for a discussion of procedures and subroutine procedures. Temporary fields are only valid for the duration of the application run, but useful for temporary results within procedures and for communication between procedures, reports, and subfiles during the processing time frame of one or more master records in the old and new work areas. Working storage fields are similar, except that the work areas involved use the W qualifier and are defined in the common library.

## Flag Fields

Work area fields referenced by an F qualifier are VISION:Builder flag fields. Flag fields are system-defined fields that provide functional and informational data in the following contexts: time, date, array processing, report pagination, transaction processing,❹ record deletions,❹end of file, and segment processing. This section emphasizes the flags that directly affect basic master file processing, namely the DELETE❹ and EOF flags.

| | |
|---|---|
| DELETE Flag❹ | Applies to VISION:Builder 4000 model series only. If this flag is set to 1 after procedures have referenced the work areas (see Procedures on page 5-8), the new work area is not written to the new master file but to the audit file,❹ if the audit file is specified by the FILE AUDIT command in the Run Control Group. |
| EOF Flag | The EOF flag is an 11-byte field that reflects the condition of each input file. Its master file condition (byte 1) is a Y until after the last master record has passed through the work areas; then it is set to E. If no old master file is specified by the FILE MASTER INPUT command in the Run Control Group, the master file condition is set to an N. Procedures can set the master file condition to an E to force a master end of file, which will terminate the run after processing the current master file record to completion. |

Because flag fields are in work areas, procedures, reports, and subfiles can reference any of them and procedures can modify some of them. The DELETE flag❹ can direct master records or segment occurrences for deletion and the EOF flag can terminate master file processing arbitrarily due to run time conditions.

For more information about flag fields, refer to the *VISION:Builder ASL Reference Guide* or the *VISION:Builder Reference Summary*.

# Application Components

A basic VISION:Builder application consists of the following components:

- Run Control
- File Definitions
- Procedures
- Reports
- Subfiles

File definitions were described in Chapter 3 and Run Control was introduced in Application Cycle Overview on page 5-3. The following sections describe in more detail the procedure, report, and subfile components of a VISION:Builder application and how these components relate to each other and to the basic master file processing cycle. See Figure 5-5 and Figure 5-6 for the general structure of how these components relate.

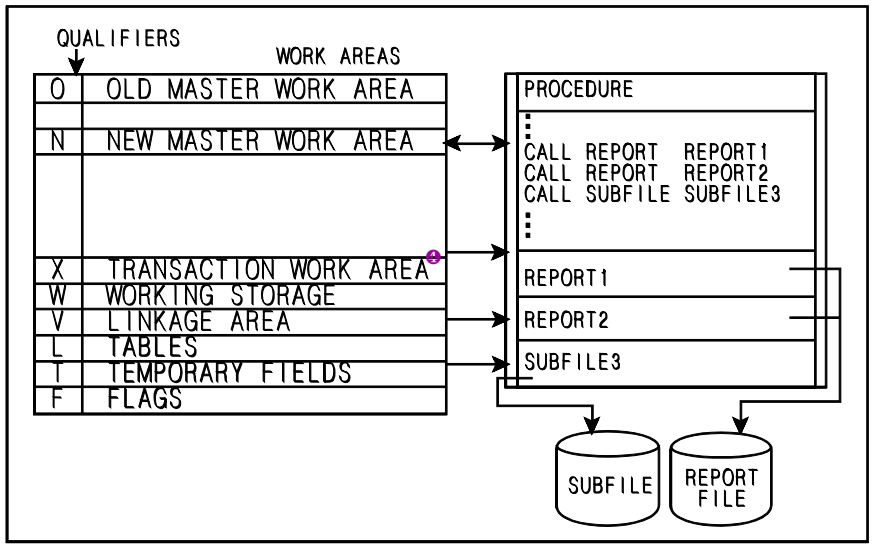



Figure 5-5 Procedure, Report, and Subfile Components\

```
CONTROL DB2 D61A INM4CALL
FILE MASTER INPUT, NAME TEMPL, KEYS NONE
FILE REPORT
FILE SUBF3, NAME OUT1
```

Figure 5-6 Run Control Identifying Old Master, Report and Extract files

## Procedures

Procedure statements reference the work areas to examine or modify data. Procedures in turn can invoke, using the CALL command, reports and subfile procedures to perform their respective output functions. Because application files pass through the work area fields, procedures can modify flag fields and direct master records or segment occurrences for deletion by appropriately setting the DELETE❹ flag. Chapter 10, *Transactions* discusses updating in detail.

Within the basic VISION:Builder master file processing cycle, there are two events that invoke procedures. The first event within the cycle is after each master file record has been read. The second event is after the master file has reached end of file. A VISION:Builder application can optionally include a unique procedure that will be invoked each time these events occur. The default is processing after each master file record is read. Other event control points are specified using the TYPE keyword on the PROC command.

```
MAIN: PROC  ; Each master file record control procedure
   (procedure logic)
END PROC
;
ENDOFILE: PROC TYPE EOF   ; End of file control procedure
   (procedure logic)
END PROC
```

Figure 5-7 Application Event Controlled Procedures

- The Each Master Record procedure (also known as the Main or type N procedure) can reference or modify the work areas while each master record occupies the old/new work areas before the old master file reaches end of file. This procedure can call subroutine procedures, reports, and subfiles.
- The End of File procedure (also known as the type EOF or E procedure) can reference or modify the work areas once after the old master file reaches end of file. It cannot reference the work area for the master file. This procedure can also call subroutine procedures, reports, and subfiles. Reports and subfiles called from this procedure can produce output that summarizes the application run.

Both of these procedures can call other procedures, known as subroutine procedures. Subroutine procedures can, in turn, call other subroutine procedures, as appropriate for good logical and structural design of the application. One restriction, however, is that procedures cannot be called recursively.

Subroutine procedures can reference or modify only the work areas that are eligible for the calling procedure. For example, a subroutine procedure called by the End of File procedure cannot reference the master file work area. The use of subroutine procedures allows you to reduce the number of redundant statements because subroutine procedures can be called by several procedures. The code within a DO block of a procedure operates as if it were a subroutine procedure with regard to rules for field references and looping.

The procedure shown in Figure 5-8 selects records from the new work area, using the N qualifier, that have a JOB value equal to PRES. Any new work area records not meeting this requirement direct VISION:Builder to branch around the remaining procedure statements, including the CALL statements for the reports and subfile. This demonstrates the selection concept.

If the temporary field COUNTER is greater than 5, VISION:Builder sets the EOF flag field to E; otherwise, VISION:Builder calls the reports and subfile. This procedure references the fields JOB and COUNTER in the new and temporary work areas using the N and T qualifiers, respectively.

```
MAIN: PROC  ; Each master file record control procedure
COUNTER: FIELD F 4 INIT 0
 IF N.JOB = 'PRES'
   LET T.TEMP1 = 'REPORT ONLY IF JOB IS PRES'
   LET T.COUNTER = T.COUNTER + 1
   IF T.COUNTER GT 5
      LET PF(F.EOF 1 1) = 'E'
   ELSE
      CALL REPORT REPORT1
      CALL REPORT REPORT2
      CALL SUBFILE SUBFILE3
   END
 END
END PROC
```

Figure 5-8 Sample Application Procedure

The procedure also modifies the fields TEMP1, COUNTER, and EOF in the new, temporary and flag work areas, respectively. The flag field EOF is only partially modified because the partial field (PF) entries force the old master file to end of file in order to shorten the application run.

## Reports

Report command groups are used to specify the content and format of the reports generated by your application. Each time a report is invoked, a record is output to a report file. In addition to the data selected by the report detail, report files contain print formatting information. The report file records are processed by a VISION:Builder report step that generates formatted reports.

The examples shown in this section are intended to illustrate how the work areas are used to obtain the report contents. Chapter 9, *Reporting* discusses the report formatting specifications. As shown in Figure 5-9, REPORT1 references the TEMP1 field from the temporary work area using the T qualifier and the EMPNO, FIRSTNME, LASTNAME fields in the new work area using the N qualifier. The values of these fields will appear on the report.

```
REPORT1: REPORT TYPE SUBROUTINE,
          ITEMS T.TEMP1, N.EMPNO, N.FIRSTNME, N.LASTNAME
END REPORT
```

Figure 5-9 Sample Report Command Group – Report #1

As shown in Figure 5-10, REPORT2 references the COUNTER and EOF fields in the temporary and flag field work areas using the T and F qualifiers, respectively.

```
REPORT2: REPORT TYPE SUBROUTINE, ITEMS T.COUNTER F.EOF
END REPORT
```

Figure 5-10 Sample Report Command Group – Report #2

Normally, all report detail definitions add report records to one report file called M4REPO. Report detail records can be interleaved as each master file record is processing in the work areas (see Figure 5-11).

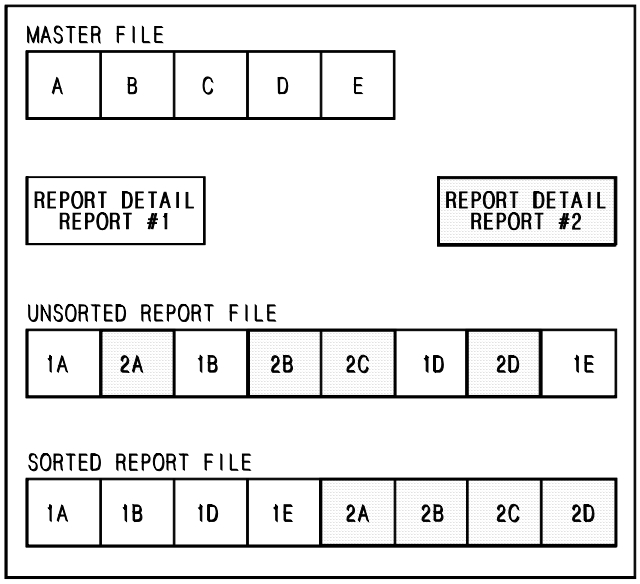



Figure 5-11 Unsorted and Sorted Report Files

The interleaved report records require a subsequent sort step that outputs a report work file, M4REPI, containing sorted report records. These records are input to a report step that produces a print image file. A processing step can produce up to nine report files — M4REPO, M4REP2, M4REP3, . . . , M4REP9.

The Report Manager function allows you to route reports and report subsets to remote sites and to collate subsets from a collection of reports and print them in a specified order.

## Subfiles

Subfiles are specified in the EXTRACT command group. A subfile record is output each time the subfile is called, allowing the procedure to be selective about which subfile records are output. In this way, your application can produce specific files as subsets of input files. In fact, subfiles can be updated versions of input files.

The application run can also provide VISION:Builder statements to catalog the subfile's data definition in the common library. These automatic definitions are optional and are derived from the present record and field definitions in the common library and temporary field definitions in procedures.

A subfile can contain selected fields from the various work areas or an entire record from an input file work area. When the EXTRACT command group specifies a set of fields, the output is an unstructured record (flat file) containing the selected fields. When the EXTRACT command group specifies an entire record from an input file work area, the subfile record contains the identical structure as the input file work area. Any structured subfile must derive its structure from an input file.

Subfiles are often produced for future processing requirements. The subfile records are not automatically sorted or ordered in any way.

The example in this section illustrates how work areas are used to specify the content of the subfile record. Additional subfile characteristics and specifications are described in Chapter 7, *Output File Processing*. As shown in Figure 5-12, SUBFILE3 references all four fields in the new, temporary, and flag field work areas that are referenced by REPORT1 and REPORT2.

```
SUBFILE3: EXTRACT FILE OUT1, TYPE SUBROUTINE,
   ITEMS T.TEMP1    T.COUNTER  N.EMPNO N.JOB,
         N.LASTNAME N.FIRSTNME F.EOF
```

Figure 5-12 Sample EXTRACT Command Group

# Basic Application Cycle

Basic master file processing combines the automatic functions of VISION:Builder and your application's procedures to process files. Figure 5-13 illustrates the processing flow of the basic application cycle. This figure is like Figure 5-3 on page 5-4 except that it also includes End of File procedures and reports.

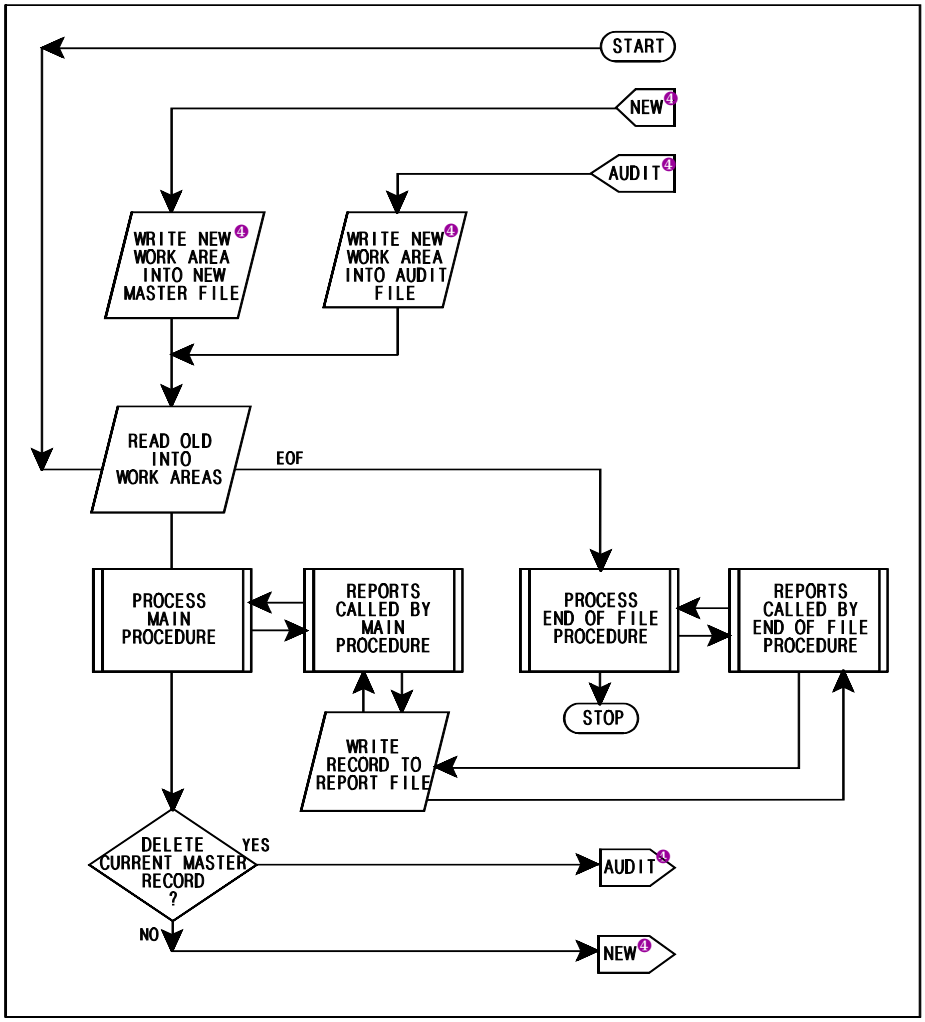



Figure 5-13 Basic Application Cycle, Including End of File Procedure

The file input/output, the work areas, and the structure of the requests are illustrated in Figure 5-14. You must associate these two view points to best understand the relationship between the automatic functions of VISION:Builder and your requests.

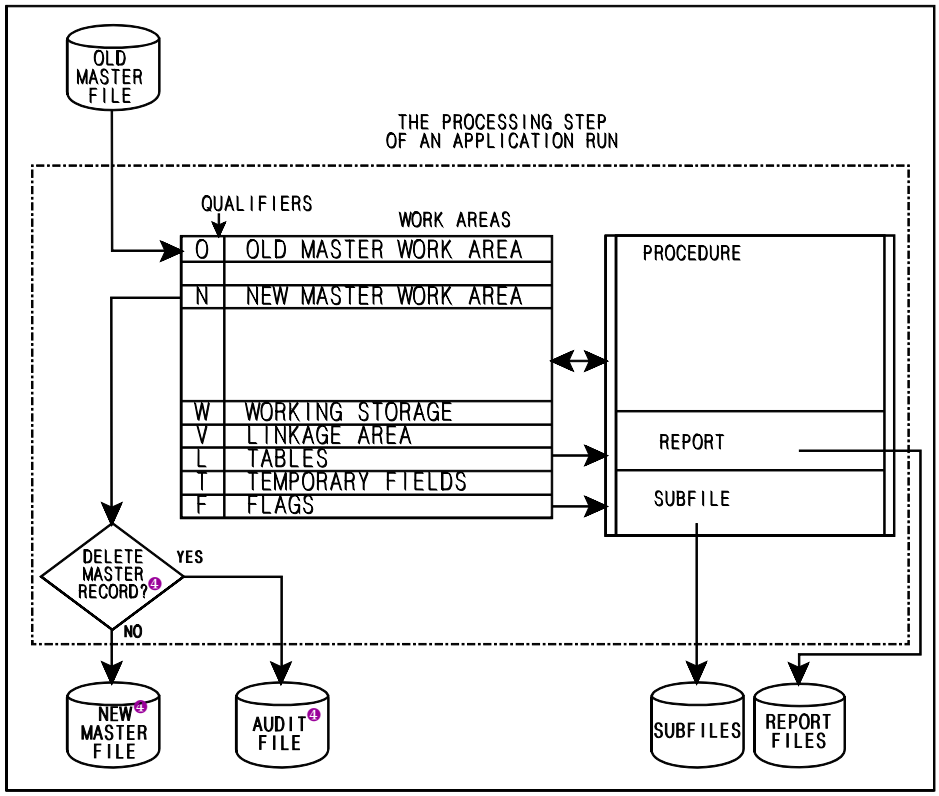



Figure 5-14 Basic Processing Step with Input and Output Files

The basic application cycle consists of the reading of the old master file record and of the invocation of the Main (Each Master Record) procedure, which in turn can call subroutine, report, or subfile components and write either the new master record❹ or an audit file record.❹ When end of file is encountered on the old master file and before the run is terminated, the End of File procedure is invoked, which in turn can call subroutine, report, or subfile components.

Processing begins when VISION:Builder reads the first old master file record in the work areas and resets the DELETE flag to zero. If the old master file is not yet at end of file, the Main procedure is invoked to reference the work areas, call subroutine procedures, reports, or subfiles, and specify the master file record or segment occurrence for deletion by replacing a 1 or 2 into the DELETE flag.❹

After each master file record is processed, VISION:Builder interrogates the DELETE flag.❹ If the DELETE flag is set to 1, no new master record is written, optionally an audit record is written, and another old master record is automatically read into the work areas. However, if the DELETE flag is set to 2, any segment type occurrence with an empty key field (blank for character fields and zero for numeric fields) is not written to the new master file and the next old master file record is read.

Within a procedure, you can choose to execute a TRANSFER TO NEXT_MASTER command. This command causes the flow to immediately branch to the point where the next old master file record is read, allowing you to conditionally control the normal processing flow. Another option available to control the normal processing flow is the EOF flag. After the processing of each master file record is completed, the EOF flag is interrogated. If the setting of the EOF flag for the old master file is E, the processing flow behaves just as if the last master file record had just been processed. The End of File procedure is invoked and the run is terminated.

## Application Cycle and Multiple Reports

VISION:Builder has the ability to generate up to 255 separate reports during a single pass of the application's input files. This is accomplished in VISION:Builder by breaking down the application into three logical steps; processing, sorting, and reporting. The processing step extracts and identifies the pieces of information for each report. The sort step organizes the pieces of information so that they are grouped by report and ordered as required within each report. The report step uses the organized pieces of information to produce each formatted report according to the application specifications.

This extended concept within the application cycle is illustrated in Figure 5-15. For the purpose of this illustration, the master file is the only input to the application. Remember, you can input up to eleven files (master, transaction, and nine coordinated files).

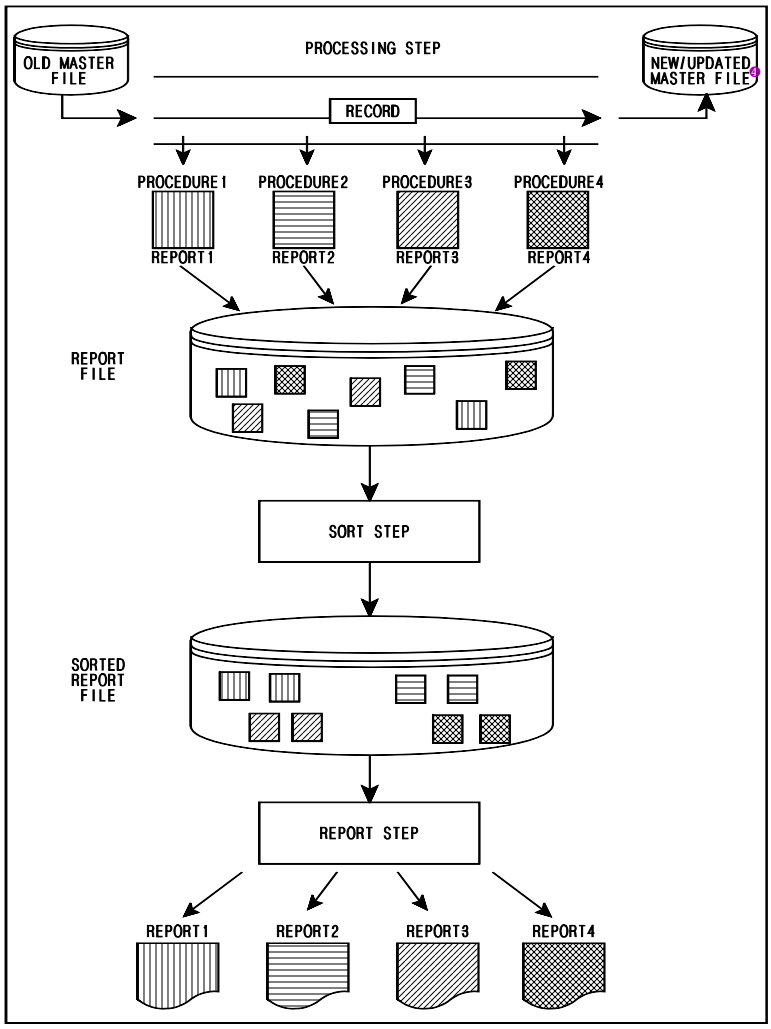



Figure 5-15 Application Cycle and Multiple Reports

During the processing step, each input record is read once into a work area that is made available to all the application procedures and reports. As the data is read, the procedure performs its algorithms and extracts the appropriate data for the required reports.

As each input record is processed by a procedure, any modifications that are made to the data in the input work area will be passed to subsequent procedures and reports.When the input record has been processed by all the procedures and reports within the application, it can be output to the new master file.❹ This option is available to applications so that updated and changed old master (input) records can be written to a new (output) master file.❹ Updates and changes can be accomplished procedurally or by using transactions.❹ See Chapter 10, *Transactions* and Chapter 15, *Advanced Transaction Processing* for information on transactions and updating.

The process of making each input record available to every procedure and report within the application is repeated for every record read from the old master. When all records have been processed from the old master file (end of file has been reached), the processing step is completed.

The data that is extracted from the input record during the processing step is placed into a transient file known as the report file. At this point, the pieces of information are just raw data. No formatting has been applied.

In the sort step, the data is grouped by report and ordered as required by the specific report.

The final step is to produce the formatted reports. The report step uses the sorted and organized report file data to generate each formatted report specified in the application.

Remember that when multiple reports are produced by an application, each input record is read once, each procedure and report extracts its appropriate information to a report file, the information is sorted, and the formatted reports are generated.

## Sort Step

A sort step is needed if a report file contains more than one report or if the only report requested is in a sequence different from the sequence of the processing step. When required, the sort step needs parameters in addition to the report file itself. These parameters are provided by VISION:Builder in the form of standard sort control statements. The installation supplied sort program arranges the report file records according to the sort control parameters. One sort step is required for each report file (M4REPO, M4REP2, M4REP3, . . . , M4REP9).

## Report Step

The report step reads the sorted report file, M4REPI, and prepares a print image file (see Figure 5-16). This process includes all of the operations pertaining to report layout, field formatting, and summarization specified on the Report Detail window (see Chapter 9, *Reporting*).

If so specified, the report step will also separate the various report outputs and summary files to their designated destinations. See Chapter 17, *Report Manager* for more information about report routing and collating. Refer to the *VISION:Builder for OS/390 Environment Guide* (OS/390 users) or the *VISION:Builder Environment Manual for VSE* (VSE users) for information about JCL requirements.

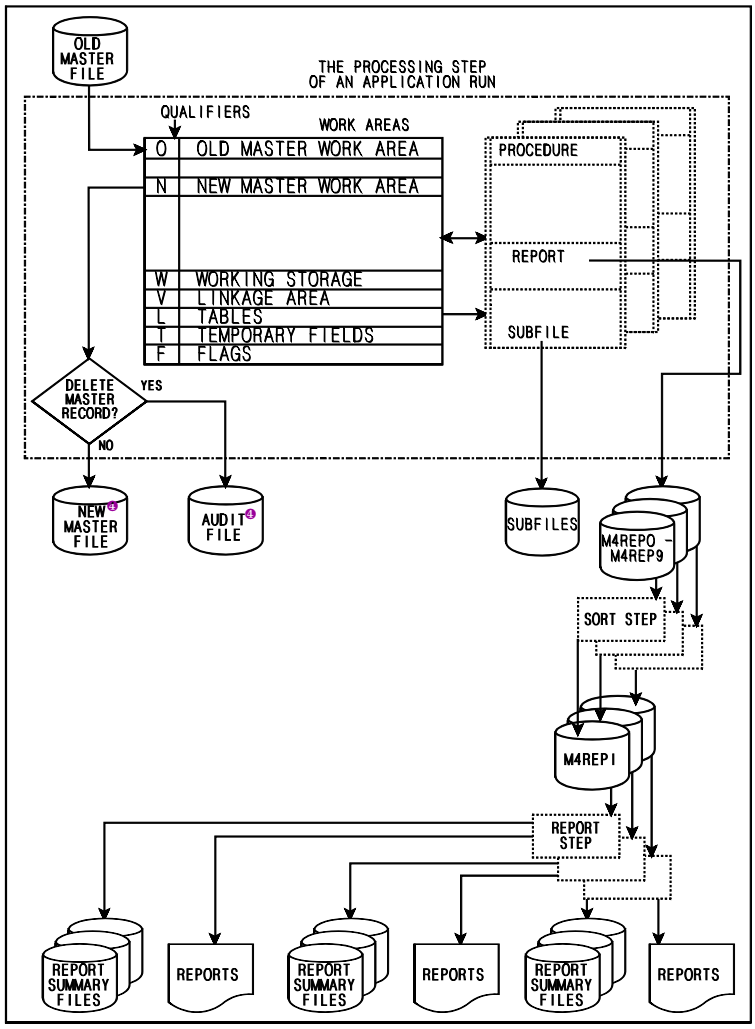



Figure 5-16 Process, Sort, and Report Steps

## Single-Step Processing

The operations performed during the application cycle described previously are sometimes referred to as the processing step. The processing step, sort step, and report step should be thought of as logical steps and not necessarily the execution of three distinct job steps in terms of the operating system job execution. When only one report file is present in an application, these three logical steps can execute as one job step, yielding a significant savings in machine resources. Refer to the *VISION:Builder for OS/390 Environment Guide* (OS/390 users) or the *VISION:Builder Environment Manual for VSE* (VSE users) for information about the considerations relating to single-step processing for your operating environment.

# Sample Application Source Listing

Chapter 2, *Application Development System* introduced the concept of an application run in which the processing for an application takes place. Figure 5-17 is the sample application used as an example in Application Components on page 5-7. A complete listing of this application with the expanded source and fixed format translation is shown in Appendix D, Figure D-1.

Note that the application begins with a Run Control command group that must always be the first statements in an application program. These statements identify the file definition for the master file and other application inputs, outputs and keywords. The statements beginning with the CONTROL command and continuing until the first procedure or request are known as the Run Control section of the application. Following the Run Control section are the various procedures, reports, and subfiles of the application. There is no required ordering of procedures, reports, and subfiles in the input stream.

If you are using fixed format syntax for preparing your application, refer to the *VISION:Builder Reference Summary* for a detail description of each statement type. If you are using Advanced Syntax Language (ASL), as shown in these examples, refer to the *VISION:Builder ASL Reference Guide* for a detailed description of the ASL statements.

```
CONTROL DB2 D61A INM4CALL LISTGEN
FILE MASTER INPUT, NAME TEMPL, KEYS NONE
FILE REPORT
FILE SUBF3, NAME OUT1
;
MAIN: PROC
; Each master file record control procedure
COUNTER: FIELD F 4 INIT 0
;
IF N.JOB = 'PRES'
   LET T.TEMP1 = 'REPORT ONLY IF JOB IS PRES'
   LET T.COUNTER = T.COUNTER + 1
   IF T.COUNTER GT 5
      LET PF(F.EOF 1 1) = 'E'
   ELSE
      CALL REPORT REPORT1
      CALL REPORT REPORT2
      CALL SUBFILE SUBFILE3
   END
END
END PROC
;
ENDOFILE: PROC TYPE EOF
; End of file control procedure
LET T.TEMP1 = 'ALL DONE'
REPORT T.TEMP1 T.COUNTER
  END REPORT
END PROC
;
REPORT1: REPORT TYPE SUBROUTINE,
                ITEMS T.TEMP1, N.EMPNO, N.FIRSTNME, N.LASTNAME
END REPORT
;
REPORT2: REPORT TYPE SUBROUTINE, ITEMS T.COUNTER F.EOF
END REPORT
;
SUBFILE3: EXTRACT FILE OUT1, TYPE SUBROUTINE,
   ITEMS T.TEMP1 T.COUNTER N.EMPNO N.JOB N.LASTNAME N.FIRSTNME F.EOF
```

Figure 5-17 Sample Application

**Note:** For the complete version of this application, see Sample Application on page D-1.

Chapter

# 6 Input File Processing

VISION:Builder automatically handles all input, output, and interfile matching of your application's data files. VISION:Builder uses number of files to do this — input and output — from which to choose, according to the requirements of your application.

This chapter describes the user input files available to VISION:Builder applications, the various types, their normal usage, and how they are processed.

You can input up to 11 data files to a VISION:Builder application. They are the old master file, a transaction file,❹ and up to nine coordinated and/or array files.

Additionally, the VISION:Builder source statement specifications are also input to all application runs, definition/maintenance runs, and common library utility runs.

The following table shows the types of input data files and their usage in a VISION:Builder program.

| VISION:Builder Name | DTF/DD Name | Usage in a VISION:Builder Application Program | Comments |
|---|---|---|---|
| Old Master | M4OLD | The primary file for the application. | M4OLD is the input master file. |
| Transaction❹ | M4TRAN | The file containing data used for updating the master file. | |
| Coordinated and/or Array | M4CORDn | File(s) containing additional input data required by the application. | n = 1 through 9 |
| Source Statements | M4INPUT | The source code that contains the application requirements in source statement format: application, definition, and utility specifications. | Input to all engine runs: definition, application, and utility. |

**Note:** This chapter contains examples to illustrate and enhance the concepts and facilities that are described. These examples are portions of complete job streams. For the complete versions of the examples, Appendix D, *Complete Sample Applications*.

The following members are referenced in this chapter:

| Job Name | Member Name | Description |
|---|---|---|
| Preparation job | R06PREPA | The Preparation job defines and builds the files and data sets used in the applications that are the examples. |
| Examples job | R06FIGSA | The Examples job contains applications that demonstrate the coding and outputs described and shown with this chapter. |

The actual JCL, source code and data for these job streams are delivered on the VISION:Builder system tape in the Examples file. This file is downloaded during the installation process. See the *VISION:Builder Installation Guide* for more information.

As you design your VISION:Builder application, keep in mind that a given data file might be designated as the old master file in one application run, the transaction file in another run, and a coordinated file in yet another run. The VISION:Builder file names only describe how the file is processed in the current VISION:Builder application. The type of data in the file and how it will be used is still application dependent.

VISION:Builder understands the format and attributes of your input data file by means of a file, transaction, or array definition previously stored in the common library. You can tell VISION:Builder which definitions to use for the input data files in your application by specifying the definition names on the FILE CORDn command or by specifying the definition names in the appropriate Run Control statements (RC, RF, and RG). You can specify the same file definition name for all input files that have the same format, structure, fields, and attributes.

VISION:Builder will perform all the file handling chores for all the access methods and database managers directly supported by VISION:Builder. In these situations, VISION:Builder performs:

- All opens and closes.
- All reads.
- All file matching where multiple input files are involved.
- Error trapping and recovery (if possible) or graceful termination.

For access methods and database management systems that are not directly supported by VISION:Builder, you can use one of the following to invoke your accessing module:

- GDBI (Generalized Database Interface)
- GSI calls
- User I/O

VISION:Builder invokes your routine each time it needs to perform an input function for the designated files. Your routine performs the requested function and passes the data to VISION:Builder in the format described in the appropriate VISION:Builder file definition. From this point, your VISION:Builder application behaves as if VISION:Builder had performed the access functions.

For more information about user I/O, see the *VISION:Builder for OS/390 Environment Guide* (OS/390 users) or the *VISION:Builder for VSE Environment Guide* (VSE users).

GDBI is described in the GDBI Programming Manual. It interacts with VISION:Builder in the same manner as User I/O, except it functions at the segment level instead of the record level.

GSI calls allow you full control and responsibility, outside of VISION:Builder's domain.

The access methods that VISION:Builder directly supports include:

- VSAM
- Sequential
- ISAM

The database management systems include:

- IMS
- DB2

There are several points to keep in mind regarding your application and how the processing flow is controlled.

- The input files and their relationships to each other essentially control the application processing run.
- The application run must have at least an old master file or a transaction file❹ as input.
- The nature of the flow through the VISION:Builder cycles and the procedures is largely controlled by the record structure and data within the master file and/or the transaction file,❹ along with the relationship of the master file with the coordinated input files.
- The collection of procedures in an application are influenced as much by the record structures and data in the input files as they are by the specific application requirements.

A vital concept behind understanding the way in which VISION:Builder processes input files is that of key uniqueness.

Key uniqueness requires that there be a set of one to nine fields within the root segment of a record that, when its values are logically concatenated, uniquely distinguishes that record from any other record in the file. VISION:Builder does not require that the group of fields whose values uniquely identify (key) the record be contiguous.

This is a common concept with database management systems and indexed or direct-access methods. In fact, many of these access methods are more restrictive with respect to the number and positioning of key fields than is VISION:Builder.

VISION:Builder does not require key uniqueness at the root level unless the access method does. However, records with equal keys can significantly affect the processing flow of a multi-file application run. Key uniqueness is especially significant for applications that use transaction-driven updating❹ or sequential coordination. In either of these situations, all the automatic file matching involves only the first equally keyed record on the file controlling the process, usually the master file.

# The Master File

The master file is the primary file for all VISION:Builder applications. The term master signifies only that the file so designated is the primary file in the application run. There is no significance to the term with respect to data or information attributes of the file. There are two work areas, old and new, that are set up to accommodate the data in the master file. The old master work area contains the data that is input to the application from the old master file and is only available when an old master file is present in the run. The new master work area initially

contains a copy of the old master data and is available for updating by the transaction input data❹ and your application procedures. The new master work area is always available to a VISION:Builder application.

The master file can be any format and structure, using any access method that can be defined to and supported by VISION:Builder. See Chapter 3, *Defining Data* for information on defining data structures. This includes hierarchically structured records from IMS databases or mapped from relational tables (for example, DB2). Note that a file of VISION:Builder array records could be input as a master file. However, the array definition cannot be used. You would need to catalog a file definition that maps the VISION:Builder array record structure. Arrays are normally input as coordinated files. Chapter 13, *Arrays* discusses arrays.

## Master File Processing Options

There are a variety of options available in VISION:Builder that control the master file processing mechanisms. The following is a list of the options, with a brief explanation, that can be used in an application for specifying how the master file should be processed:

- Sequencing Checking on page 6-5
- Sequential or Direct-Access on page 6-5
- Update-in-Place on page 6-6
- Start Search and End Search on page 6-6
- One Buffer on page 6-7
- MOSAIC on page 6-8
- Preselection on page 6-8
- WH Statement on page 6-8

Normally, VISION:Builder processes the entire old master input file in read-only mode, sequentially, and performs an ascending sequence check of key(s) in the root segment, looking for duplicates. This is the automatic default processing mode, but it can be customized to accommodate your specific application requirements.

### Sequencing Checking

You can have the sequence checking performed but allow for duplicates or you can bypass all sequence checking entirely. You use the FILE CORDn command or the Run Control statement (RC) to specify your sequence checking options. See Figure 6-1 for an example of a sequence checking specification on the FILE CORDn command. Remember that your choice of a sequence checking option will have an impact on the coordination and transaction processing❹ mechanisms.

### Sequential or Direct-Access❹

You have an option that allows you to access the master file directly instead of sequentially. Sequence checking is not performed. The master file type and access method must support random access and direct-reads. In this mode of operation, a transaction file must exist in the application and only master file records that

have transactions are accessed. You use the FILE CORDn command or the Run Control statement (RC) to specify the direct-read option. See Figure 6-1 on page 6-6 for an example of the direct-read specification on the FILE CORDn command.

## Update-in-Place❹

**Note:** Update-in-place will allow you to create an application that will sequentially read an old master record, procedurally change the data in the new master work area, and rewrite the updated record back to the file without using transactions.

When you specify update-in-place for the master file, you rewrite the updated new master work area back to same physical location that contained the original old master input record. In Chapter 7, *Output File Processing*, the option to output the updated new master work area to an entirely new data set is described. A master file can be updated-in-place only if the storage medium is a direct-access storage device. Update-in-place can be specified for direct or sequentially accessed masters.

The FILE MASTER command is used to specify processing options for the master file in an application. Here is a sample showing some master file processing options:

```
FILE MASTER UPDATE,       1
     definition-name,
     ACCESS DIRECT,       2
     KEYS EQUAL           3
```

Figure 6-1 Sequence Checking, Direct-Access, and Update-in-Place on the File Master Command

The following notes refer to the matching numbered lines in Figure 6-1:

| | | |
|---|---|---|
| **1** | `MASTER UPDATE❹` | Specifies update-in-place. |
| **2** | `ACCESS DIRECT❹` | Specifies direct-reads on the old master file. |
| **3** | `KEYS EQUAL` | Specifies duplicate keys on the old master file. |

See the *VISION:Builder Reference Summary*, RC statement position [23], for a chart on the combinations of direct-read with update-in-place and their impact on other application activities.

When you specify update-in-place, the old master file is processed in read/write (update) mode. Usually, the old master file is processed as read-only.

## Start Search and End Search

The Start Search and End Search options are used to limit the number of records accessed in the old master file. Only records with a key value that is equal to or greater than the start search value and equal to or less than the end search value are processed. When the start value is blank, processing starts with the first record in the file. When the end value is blank, processing continues until end of file is reached. These options are only valid when the old master file exists in the

application run and is being processed sequentially. VISION:Builder assumes that the master file keys are in ascending order. If the file is not in ascending order, it is possible that all the records with keys that are within the start/end search range will not be processed.

When an old master file that is using a sequential access method is processed with these options, all records in the file are read until a key value greater than the end search value is encountered, causing a logical end of file to occur. The records in the old master file with keys less than the start search value are read, but they are bypassed and not available to the application.

When an old master file that is using a direct-access method❹ is sequentially processed using these options, a direct-read is used for the first read. This positions the read processing mechanism to the first record in the file with a key equal to or greater than the start search value. Sequential processing continues from that point until a key with a value greater than the end search value is encountered, causing a logical end of file to occur.

The start search and end search values are specified on the FILE MASTER command or in the Run Control statement (RC). See Figure 6-2 on page 6-8 for an example of the FILE MASTER command with start search and end search entries.

### One Buffer

The One Buffer option allows you to control the allocation of work areas associated with the master file. Using this option causes VISION:Builder to allocate only one buffer/work area, for the master file.

Normally, VISION:Builder allocates two work areas, old and new, for handling the processing of the master file. For applications that perform updating❹ of the master file, the two work areas are necessary for internal processing mechanisms and data integrity. However, in retrieval-only applications, there may not be a need to maintain two work areas for the master file.

You can specify the One Buffer option on the FILE MASTER command or by including the appropriate Run Parameter statement (RP) as part of your application's run control group. See Figure 6-2 for an example of the FILE MASTER command and the One Buffer entry.

There are some considerations when using the One Buffer option.

- This option is only valid in applications that access the old master in retrieval only mode. The use of transactions❹ or update-in-place❹ in the application causes VISION:Builder to ignore the One Buffer specification.
- References to fields in the old and/or new master work areas, using the appropriate qualifier, now refer to the same single common record work area (that is, there is only one single location where the master file data is stored).
- Simultaneous looping on the old and new master work areas does not occur. See Chapter 8, *Data Structure Handling* regarding looping and structured records.
- Using this option reduces the amount of storage required for the master file work areas. Application performance improves because the record data is not copied to the new work area from the old work area during the read operation.

### MOSAIC

This option controls the processing of the master file. It has impact on how segments of the master file structure are accessed through the database managers that handle IMS, relational tables (for example, DB2), and GDBI structures. See Chapter 16, *Database Access Methodology* for more details on the MOSAIC (memory optimized) option.

### Preselection

This option is used by coding a procedure that qualifies the segment access calls used to process IMS databases. See Chapter 16, *Database Access Methodology* for more details on preselection procedures.

### WH Statement

This option is used by coding a DB2 WHERE clause on a VISION:Builder WH statement and including it as part of the application's run control group. The WH statement is used to control the processing of relational tables (for example, DB2). See Chapter 16, *Database Access Methodology* for details on the WH statement in conjunction with how VISION:Builder constructs DB2 statements. See Appendix A, *Sample DB2 Applications* for details on including WH statements in an application.

Here is a sample showing more master file processing options:

```
FILE MASTER INPUT,
     definition-name,
     STARTKEY 'key-value',                              1
     ENDKEY 'key-value',                                2
     MOSAIC,                                            3
     ONEBUFFER,                                         4
     SEGMENT segment-name WHERE "sql-where-clause",     5
     SEGMENT segment-name SSA "pre-selection-ssa"       6
```

Figure 6-2 More FILE MASTER Command Parameters

Note that the coding example is for reference purposes only and there are conflicting options shown (see the *VISION:Builder ASL Reference Guide* for details.) For instance, the SEGMENT keyword phases shown indicate a WHERE clause for Relational Table Input and the SSA clause for IMS Data Base Input.

The following notes refer to the matching numbered lines in Figure 6-2:

| | | |
|---|---|---|
| 1 | `STARTKEY` | Processes old masters with keys ≥ A0001. |
| 2 | `ENDKEY` | Processes old masters with keys ≤ A9999. |
| 3 | `MOSAIC` | Specifies that this processing option be used for the master file. |
| 4 | `ONEBUFFER` | Specifies that only one old master work area is needed. |
| 5 | `SEGMENT WHERE` | Specifies the SQL WHERE Clause. |
| 6 | `SEGMENT SSA` | Specifies the IMS Preselection Qualifications. |

## The Transaction File❹

The transaction file consists of records that are used to add, delete, or modify data on a master file. The identification of and the actions that individual records perform are defined in the transaction definition for the particular master file. A file definition that maps the transaction record structure can also be used to access the transaction work area during procedure and request processing. See Chapter 10, *Transactions* and Chapter 15, *Advanced Transaction Processing* for details on transaction definitions and transaction processing, respectively.

The transaction file is always read sequentially, while the master file can be read sequentially or directly (the default is sequential). When the master file is read directly, it is read in the order that transaction record keys match the master file record key(s).

The transaction file can be specified as any format, structure, and access method, except IMS database, relational (DB2) table, or GDBI. The transaction file record can be hierarchically structured. However, only the root segment (fixed portion) of the record can be referenced in the transaction definition. The lower level segments in a structured transaction file can only be used for procedural updating, processing, and reporting. A file of VISION:Builder array records could be used as a transaction file; however, the array definition cannot be used to map the file. To do so, you would need to catalog a file and transaction definitions that map the VISION:Builder array record structure.

Remember that you can use the transaction file and a transaction definition in an application that builds a temporary new master file structure in memory that is only used to produce your required outputs. See Chapter 15, *Advanced Transaction Processing* for additional information.

# Coordinated Files

Coordinated files are another type of input file that can be processed by a VISION:Builder application. You use coordinated files when your application requires that data from different files to be brought together. A VISION:Builder application run can include up to nine coordinated files.

Coordinated files can be any format and structure, using any access method that can be defined to and supported by VISION:Builder. This includes hierarchically structured records from IMS databases or mapped from relational tables (for example, DB2). You can also input files that contain VISION:Builder array records as coordinated files using the array definition. Coordinated input files are the only input files that can use an array definition as the file definition for application processing.

These input files are called coordinated files because VISION:Builder automatically coordinates the reading and alignment of these files based on the type of coordination mechanism being used in the application. You only need to choose the coordination method that satisfies your application processing requirements.

The reading and processing of coordinated files is controlled by one of three coordination mechanisms:

- Indexed Coordination. VISION:Builder automatically reads the coordinated file directly and randomly based on a value in any field from any work area available to your application. See Indexed Coordination on page 6-11 for the details.
- Sequential Coordination. VISION:Builder automatically synchronizes the master and coordinated input files by reading the files sequentially and matching their key fields. See Sequential Coordination on page 6-17 for details.
- User Coordination. VISION:Builder reads a coordinated input file serially, directly, or both when procedurally instructed by the application logic. The timing and synchronizing is under the control of the application's logic. See User Coordination on page 6-46 for the details.

Here are some general points regarding coordinated files.

- The generic DTF/ddname for coordinated files is M4CORDn. The substitution of a digit for “n” identifies a specific coordinated file. This assignment is made on the FILE CORDn command or in the Additional Files (RF) Run Control statement.
- Fields in a coordinated file are referenced in procedure and request statements by using the number of the coordinated file as a qualifier. For example, a field name with a qualifier of 2 comes from M4CORD2.
- The VISION:Builder flag field ECORD (described below) is 9 bytes long. One byte is reserved for each potential coordinated file. The key matching status of M4CORD4 is flagged in the fourth byte of the ECORD flag field.

You must include a FILE CORDn command or an Additional File (RF) statement for each coordinated file input to your application run. This statement specifies the presence of a coordinated file, as well as the type of coordination used for its processing. With the exception of a run involving multi-file sequential coordination (Sequential Coordination on page 6-17), it is unimportant which file you assign to M4CORD1 and which to M4CORD2. Numbers can be skipped. See the *VISION:Builder ASL Reference Guide* for more information about entering information on the FILE CORDn command.

### ECORD Flag

The flag field ECORD communicates to procedures the status of coordinated files' key matching. The ECORD flag field is nine bytes long. The third byte of the ECORD flag field communicates the status of M4CORD3.

Because the use of the ECORD flag varies slightly by type of coordination, each of the following subsections will include more specifics about its use.

**RULE:** When referencing the ECORD flag, always include a partial field specification to isolate the byte of the ECORD flag field corresponding to the specific coordinated file. VISION:Builder syntax checking does not enforce this rule.

## Indexed Coordination

Indexed coordinated files are read directly (and randomly) by VISION:Builder using the key of the record. Indexed coordinated files can be any format and structure that is supported by VISION:Builder. This includes hierarchically structured records from IMS databases or mapped from relational tables (for example, DB2). You can also input files that contain VISION:Builder array records as indexed coordinated files using the array definition. You can use indexed coordination when your indexed coordinated file's (ICF's) access method allows direct or random access by a key value. VISION:Builder allows up to nine indexed coordinated files in an application run. VISION:Builder automatically supports the following file types.

- DB2 Relational Tables
- IMS Databases
- Key Sequenced VSAM
- Entry Sequenced VSAM (if accessed through an alternate index)
- ISAM

GDBI provides the best method for accessing data from other database management systems or access methods. See the GDBI Programming Manual for more information.

The User I/O facility will also allow you to use any other direct or random access method or other database management systems and/or files. Refer to the *VISION:Builder for OS/390 Environment Guide* (OS/390 users) or the *VISION:Builder Environment Manual for VSE* (VSE users) for information.

On the FILE CORDn command in the DIRECT BY entry for the indexed coordinated file, specify the coordinating field that will contain the key value of the record in the coordinated file. Each time a procedure or request in your application executes a reference to any field in the indexed coordinated file, VISION:Builder intervenes just prior to the execution of that statement. Using the value in the coordinating field, VISION:Builder causes the appropriate record to be accessed and places the contents of that record in the work area for the indexed coordinated file.

The value in the coordinating field for the indexed coordinated file is used to access a record from the file if that record is not already in the work area. This field can be in any other file work area or in any temporary, working storage, or linkage section field. The ICF coordinating field cannot be an automatic table lookup result field.

The coordinating field can be a field from another indexed coordinated file; however, VISION:Builder does not allow a circular chain of indexed coordinated files. For example, if the coordinating field for M4CORD2 is in M4CORD1, the coordinating field for M4CORD1 cannot be in M4CORD2.

Specify the name and location of the coordinating field on the FILE CORDn command. In the DIRECT BY keyword, specify the name and qualifier of the coordinating field.

Fields from an indexed coordinated file can be referenced in all types of procedures. Immediately prior to the execution of such a reference, VISION:Builder will attempt coordination.

- VISION:Builder validates the coordinating field. If it is missing or invalid, the coordinated record work area is marked unavailable and no read is attempted. The ECORD byte for the coordinated file is set to X and control is returned to the request that triggered the coordination attempt.
- If the key of the coordinated file and coordinating fields have different properties (type, length, or scale), the coordinating field is converted to the properties of the coordinated file's record key. If the conversion is not valid, the file's ECORD byte is set to X, the file's work area is marked unavailable, and control is returned to the procedure or request.
- When the coordinated field is part of a lower level segment, a loop is started. VISION:Builder checks the status of the current coordinated file work area. If a loop is in progress on the current record in the work area or if a FIND function (FS operator) is active for a segment of the record, no read will be attempted even if the key of the coordinated file record does not match the value of the coordinating field. The value of the ECORD byte for the coordinated file is not changed. Control is returned to the procedure or request that initiated the coordination attempt.
- If the coordinated work area is currently marked unavailable, VISION:Builder checks the contents of the coordinating field against the key value of the last coordination attempt. If the coordinating field has not changed, the read is not

attempted again. The ECORD byte for the coordinated file is set to X and control is returned to the procedure or request that initiated the coordination attempt.

- VISION:Builder compares the coordinating field's value to the key of the coordinated file record that is currently in the active work area. If the two fields are equal, the coordination attempt is complete. The ECORD byte for the coordinated file is set to M and control is returned to the procedure or request that triggered the attempt.
- Using the appropriate access method, VISION:Builder attempts to retrieve a record with a key value equal to the current contents of the coordinating field.
  - If the retrieval is successful, the record contents are moved to the coordinated file's work area. The ECORD byte for the coordinated file is set to M and control is returned to the procedure or request that triggered the attempt.
  - If the retrieval is unsuccessful, the coordinated file's work area is marked unavailable. The ECORD byte for the coordinated file is set to X and control is returned to the procedure or request that triggered the attempt.

At the conclusion of the indexed coordination attempt, the initiating procedure or request resumes processing, beginning with the statement that triggered the coordination attempt.

### Indexed Coordination Example

The nature of indexed coordination is such that the indexed coordinated file can be thought of as an extension of the data segment that contains the coordinating field. For example, a customer file includes a segment of ordered item (ITEMORD) information but this segment does not contain the item name. To print the item name on a report, assign the customer file to the VISION:Builder master file and use indexed coordination to retrieve the item name from the root segment of the inventory file. Figure 6-3 through Figure 6-6 illustrate this example.

Figure 6-3 shows the structural schematic of two files.

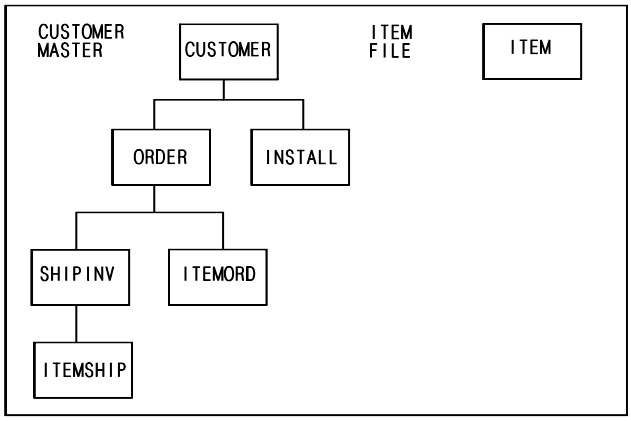



Figure 6-3 Structured Schematic of Customer Master and Item File

Figure 6-4 shows the shipping list reports prepared for new customer orders.

11/03/01 SHIPPING LIST REPORT - ITEMS ORDERED PER CUSTOMER PAGE 1
DIRECT READ ITEM FILE FOR NAME

| CUSTOMER NUMBER | ORDER NUMBER | STOCK NO. ORDERED | ITEM NAME | ORDER QUANTITY |
|---|---|---|---|---|
| 00002 | 20103 | BL-H11A | BLDR HANDBOOK ASL | 3 |
| | 20105 | BL-H11B | BLDR HANDBOOK FIXED | 3 |
| 00002 | | | | 6 |
| 00003 | 30101 | BL-B01B | BLDR QUICK REF | 5 |
| 00003 | | | | 5 |
| 00010 | 01101 | BL-T01B | BLDR TRAINING BASIC | 1 |
| | 01104 | BL-T02B | BLDR TRAINING ADVAN | 1 |
| 00010 | | | | 2 |

11/03/01 SHIPPING LIST REPORT - TOTAL WEIGHT PER CUSTOMER PAGE 1
DIRECT READ ITEM FILE FOR NAME AND WEIGHT

| CUSTOMER NUMBER | ORDER NUMBER | STOCK NO. ORDERED | ITEM NAME | ITEM WEIGH | ORDER QUANTITY | SHIPPING WEIGHT |
|---|---|---|---|---|---|---|
| 00002 | 20103 | BL-H11A | BLDR HANDBOOK ASL | 3.25 | 3 | 9.75 |
| | 20105 | BL-H11B | BLDR HANDBOOK FIXED | 3.75 | 3 | 11.25 |
| 00002 | | | | | 6 | 21.00 |
| 00003 | 30101 | BL-B01B | BLDR QUICK REF | .75 | 5 | 3.75 |
| 00003 | | | | | 5 | 3.75 |
| 00010 | 01101 | BL-T01B | BLDR TRAINING BASIC | 17.75 | 1 | 17.75 |
| | 01104 | BL-T02B | BLDR TRAINING ADVAN | 23.75 | 1 | 23.75 |
| 00010 | | | | | 2 | 41.50 |

Figure 6-4 Customer Order Shipping List Reports

Figure 6-5 shows the Run Control commands that include the customer and inventory files.

```
CONTROL NAME F144 TERM LISTGEN
FILE MASTER INPUT  CUSTOMER
FILE CORD1  ITEMFILE DIRECT BY O.ITEMORD     1
FILE REPORT
```

Figure 6-5 Indexed Coordination to Produce Shipping Lists: The Run Control Commands

**Note:** For a complete version of this example, see Indexed Coordination to Produce Shipping Lists: The Run Control Commands – Part 1 of 5 on page D-32.

The following note refers to the matching numbered line in Figure 6-5.

**1** The Direct By Clause entry identifies the file where the coordinating field (the Alternate Key) resides and signals VISION:Builder that this coordinated file (M4CORD1) is an indexed coordinated file. This statement specifies that the coordinating field is named ITEMORD and this field is in the old master (letter O) file.

```
RPTITEMS: REPORT O.CUSTNO O.ORDERNO O.ITEMORD,                    1
                 1.ITEMNAME O.ITMQTYOR,                           2
        TYPE SUBROUTINE
        ORDER BY O.CUSTNO
        GROUP BY O.CUSTNO
        TOTAL O.ITMQTYOR BY O.CUSTNO
        TITLE 'SHIPPING LIST REPORT - ITEMS ORDERED PER CUSTOMER'
        TITLE 'DIRECT READ ITEM FILE FOR NAME'
        FORMAT SUMMARYLABELS SUPPRESS
        END REPORT
```

Figure 6-6 Indexed Coordination to Produce Shipping List Report 1: The REPORT Command and Keywords

**Note:** For a complete version of this example, see Indexed Coordination to Produce Shipping Lists: Report 1 REPORT Commands – Part 3 of 5 on page D-33.

The following notes refer to the matching numbered areas in Figure 6-6.

**1** `ITEMORD` The fact that the coordinating field (ITEMORD) is referenced is entirely coincidental to the process of indexed coordination. The preceding discussion of how indexed coordination works would not change if you removed the reference to O.ITEMORD. Furthermore, the process just described would remain nearly the same if the report had no information from the ITEMORD segment. If this were the case, the first reference to a field in the indexed coordinated file would have executed an implicit reference to the coordinating field ITEMORD. Thereby, loops would be initiated at that statement through all occurrences of the ITEMORD segment for the current ORDER segment. See Chapter 8, *Data Structure Handling* to review VISION:Builder looping.

**2** `ITEMNAME` Because this statement references a field (ITEMNAME) in the indexed coordinated file (M4CORD1), VISION:Builder performs an indexed coordination attempt before executing this statement. A successful attempt (retrieval of an ITEM record from the coordinated file) makes the ITEMNAME from the coordinated file available for reporting (or any other request processing). Conceptually, a root segment from the ITEM file has been appended to an ITEMORD segment in the CUSTOMER file.

```
RPTWEIGH: REPORT O.CUSTNO O.ORDERNO O.ITEMORD,
                    1.ITEMNAME 1.ITMWEIGH O.ITMQTYOR T.SHPWEIGH,
       TYPE SUBROUTINE
       ORDER BY O.CUSTNO
       GROUP BY O.CUSTNO
       TOTAL O.ITMQTYOR BY O.CUSTNO
       TOTAL T.SHPWEIGH BY O.CUSTNO
       TITLE 'SHIPPING LIST REPORT - TOTAL WEIGHT PER CUSTOMER'
       TITLE 'DIRECT READ ITEM FILE FOR NAME AND WEIGHT'
       FORMAT SUMMARYLABELS SUPPRESS
       END REPORT
```

Figure 6-7 Indexed Coordination to Produce Shipping List Report 2: The Report Command and Keywords

**Note:** For a complete version of this example, see Indexed Coordination to Produce Shipping Lists: Report 2 REPORT Commands – Part 4 of 5 on page D-34.

**RULE:** When the indexed coordinated file's coordinating field is in a lower level segment of a hierarchical data structure or in an array cell, the first reference to a field in the coordinated file is also an implicit reference to the coordinating field. If a previously executed statement has not initiated the appropriate data structure driven loops, this implicit reference will do so.

If your coordinating field is in a lower level segment or in an array cell and your request initiates segment or array cell driven loops in the indexed coordinated work area, be aware that you are dealing with multi-path looping (see Chapter 8, *Data Structure Handling*). A multi-path nested loop performs as you would expect.

**RULE:** If you want to specify multi-path parallel looping or to use a FIND function (FS operator) with the indexed coordinated file, keep in mind that if a loop or FIND function (FS operator) is active within an indexed coordinated file work area, no coordination attempts are performed.

If you are using the FIND function (FS operator) with the indexed coordinated file, the implication of this rule is quite simple: execute a RELEASE command (RS operator) prior to the next coordination attempt. If you need the FIND function (FS) to be active for reporting, perform the reporting with a subroutine procedure or request and release the segment using the CALL command upon return from the subroutine procedure or request.

The implication of the above rule is more serious, however, if you specify multi-path parallel looping. Because the outer loop is activated first, VISION:Builder moves to the next occurrence of the segment or cell, containing the coordinating field before exhausting the parallel loop in the coordinated file. The preceding rule causes VISION:Builder to make no further coordination attempts for the duration of the procedure or request. Review the discussion of parallel looping in Chapter 8, *Data Structure Handling*.

**RULE:** If the indexed coordinated file's coordinating field is in a lower level segment or in an array cell and the coordinated file is structured or an array, do not specify multi-path parallel looping.

## Sequential Coordination

Sequential coordination automatically synchronizes a sequentially accessed master file with sequentially accessed coordinated input files. This sequential file coordination cycle is based on the classic low, equal, high matching logic. See Figure 6-8 for a simplified overview of the sequential file coordination cycle.

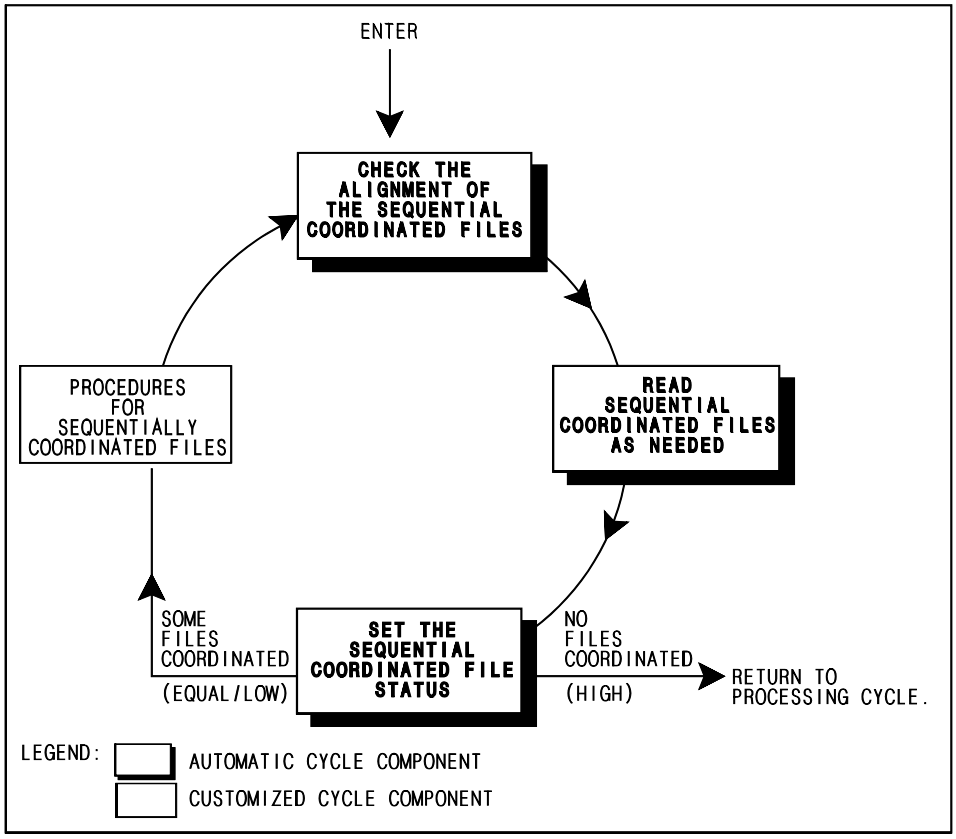



Figure 6-8 Simplified Overview of the Sequential Coordination Cycle

VISION:Builder automatically checks the alignment of the sequential master and the sequentially coordinated input files to determine their relationships. The alignment is based on each file's record keys. A read is performed on the appropriate sequentially coordinated files to get the next records into the input work area. VISION:Builder sets the ECORD flag byte of each sequentially coordinated file to reflect its relationship and status with the master and other sequentially coordinated files.

When coordination has taken place, equal and/or low relationships exist; the application's procedures that will use this status information and the available input file data are given control so that they can perform their algorithms and produce any output. When the master and the sequentially coordinated input files are no longer coordinated, all matching relationships are high; the sequential file coordination cycle is complete for the current master record. The application continues with the standard processing cycle.

There are several variations and processing options that are available for tailoring the sequential file coordination cycle to an application's requirements. These variations and options are described in this chapter.

Sequentially coordinated files can be any format and structure, using any access method that can be defined to and supported by VISION:Builder. This includes hierarchically structured records from IMS databases or mapped from relational tables (for example, DB2). You can also input files that contain VISION:Builder array records as coordinated files using the array definition.

Remember that any database management system or access method that is not directly supported by VISION:Builder can be integrated using GDBI or User I/O.

To use sequential coordination, the files that are matched to each other, including the master file, must have a common key value and must be accessible sequentially by key. All of the files involved in sequential coordination must be sequenced alike (by the key values that they share in common). VISION:Builder assumes such sequencing. VISION:Builder uses the common key values to determine which records from which files constitute a coordinated (matching) set.

You can include other types of coordinated files (such as indexed, user) in an application run that also includes sequentially coordinated files. The coordinating field for an indexed coordinated file can even reside in a sequentially coordinated file. The processing of these other types of coordinated files is, however, entirely separate from sequential coordination. The remainder of this chapter assumes application runs where only sequential coordination is involved.

You can include up to nine coordinated files in your application run. Sequentially coordinated files matching the master file are always matched to the new master work area. A sequentially coordinated file can match to another sequentially coordinated file.

A match file is any sequential file against which one or more sequentially coordinated files are matched. The application's master file is always a match file. A sequentially coordinated file can also be a match file in its relationship to another sequentially coordinated file.

The match key is the field that exists in the match file that VISION:Builder uses to determine whether or not two files' records are in a matching condition.

The match key for a match file is composed of all or part of the key fields specified for that file in its VISION:Builder file definition. The match key for a sequentially coordinated file is composed of all or part of the key fields specified in the file definition, unless you provide an overriding set of coordinating fields on the FILE CORDn command KEYNAME keyword using the Additional Files (RF) statement. Match keys must be in the root segment of a file. The coordinating file should be sequenced according to this key.

When two files are coordinated, each coordinated file key is converted to a data type and scale that is compatible with the corresponding match file key and individually compared. The first coordinated file key always corresponds to the

first (major) key of the match file. If the number of coordinating fields (key fields) differs between the two files, the file with the fewest number of key fields determines which fields are used for the comparison.

**RULE:** With sequential coordination, VISION:Builder advances all coordinated files until a non-matching record is found, while the initial matching record in the match file is retained in its work area. All coordinated file records with equal keys match to the first record on the match file with that same key.

Therefore, if a match file contains more than one record for a match key value, the second and subsequent records are processed only after the files that coordinated to that match file are moved to a non-matching condition. Remember this rule when you are deciding which files to designate as match file(s). If all of your files have duplicate records, see Chapter 15, *Advanced Transaction Processing* for a possible solution using transactions to rebuild your master file.

**Note:** If the match key contains fewer fields than the full record key as specified in the file definition, a duplicate record condition could occur.

Essentially, VISION:Builder collects a set of records (one from each sequentially coordinated file) that is low, equal, or high to the match key. VISION:Builder sets the appropriate ECORD bytes to reflect the status of each file (M for match or equal, L for low, or X for no match or high) and returns this set of records to sequentially coordinated procedures for processing.

A sequentially coordinated procedure or request includes references to at least one sequentially coordinated file field or to the ECORD flag. Sequentially coordinated procedures or requests are executed when any byte of ECORD contains the settings of M or L. This process continues until all sequentially coordinated files are moved to a key high position relative to the master file; all ECORD byte(s) are set to X.

When all sequentially coordinated files are advanced to the key high condition, all ECORD bytes are set to X and the Main (Normal) procedures are processed one additional time. This is called the follow-up pass. The follow-up pass is often useful for processing summary information collected in sequentially coordinated procedures while processing matching coordinated file records. However, all of the ECORD bytes contain X during the follow-up pass, just as they would in the situation where there are no matching coordinated file records for a given master file record.

A system flag field called MISSPASS is available for additional testing of sequential coordination status during the follow-up pass. The MISSPASS flag is a 1-byte character field. Its value is automatically set by VISION:Builder at the start of the follow-up pass. The X setting means that no matches occurred during sequential coordination for the master record; an M indicates that at least one match occurred during sequential coordination for the master record. You can test this flag in your procedures. During the follow-up pass, while executing your main procedures, F.MISSPASS EQ M selects those master file records that had at least one match during sequential coordination. F.MISSPASS EQ X selects those master file records that did not have matches during sequential coordination.

While the ECORD flag indicates the status of sequentially coordinated files, the MISSPASS flag indicates the status of the master file. Reference to the MISSPASS flag does not make a procedure sequentially coordinated. The MISSPASS flag is blank when the follow-up pass is not in progress.

If you want to skip the follow-up pass, specify the CORDONLY keyword on the CONTROL command, as shown in Figure 6-9. Sequentially coordinated procedures only process master and coordinated file records that match (ECORD setting of M) and/or, if unmatched coordination is specified, those coordinated records that are low to the master record (ECORD setting of L). It is important to note that those procedures that *are not* sequentially coordinated do not execute when the CORDONLY keyword is specified on the CONTROL command. To skip the extra follow-up pass when using fixed format syntax coding, include a Run Parameter (RP) statement with CORDONLY specified as part of your run control group.

```
CONTROL NAME run-name,
        CORDONLY
```

Figure 6-9 Skip Extra Coordination: CORDONLY Keyword on the CONTROL Command

To terminate the sequential file coordination cycle for a particular master file record, execute TRANSFER TO NEXT_MASTER. VISION:Builder immediately terminates the sequential file coordination cycle and the follow-up pass does not occur.

If you are already in the middle of the follow-up pass, a TRANSFER TO NEXT_MASTER terminates processing for the current master file record immediately. Any number of procedures that are not sequentially coordinated can be bypassed for that particular master file record.

When you terminate processing of the current master record, coordinated file records (including those currently in the work areas) are bypassed for the current master file record. The match process resumes on the next master file record read.

When the master file and all sequential coordinated files are at end of file, the data in their respective work areas is not available for processing. When the End of File (type E) procedures are executed, any reference to master file or sequentially coordinated file fields causes invalid results and missing output. This includes any subroutine procedures or requests that might be called. During the execution of the End of File procedures, the EOF flag settings for the master file and all sequentially coordinated files are E; the ECORD flag settings are X for all the sequentially coordinated files; and the MISSPASS flag is set to a value of blank.

## Single File Sequential Coordination

Single file sequential coordination can be done two ways:

- Process only those records in the coordinated file that match to a record on the master file.
- Process all records in the coordinated file whether or not they match to a record on the master file.

If you specify the MATCHONLY keyword on the FILE CORDn command (RF statement), the application processes only matching coordinated file records. If you specify the ALLRECS keyword (for unmatched), the application processes all coordinated file records.

## Matched Record Only Processing

Figure 6-10 illustrates the flow of matched record only processing for a master file and one sequentially coordinated file.

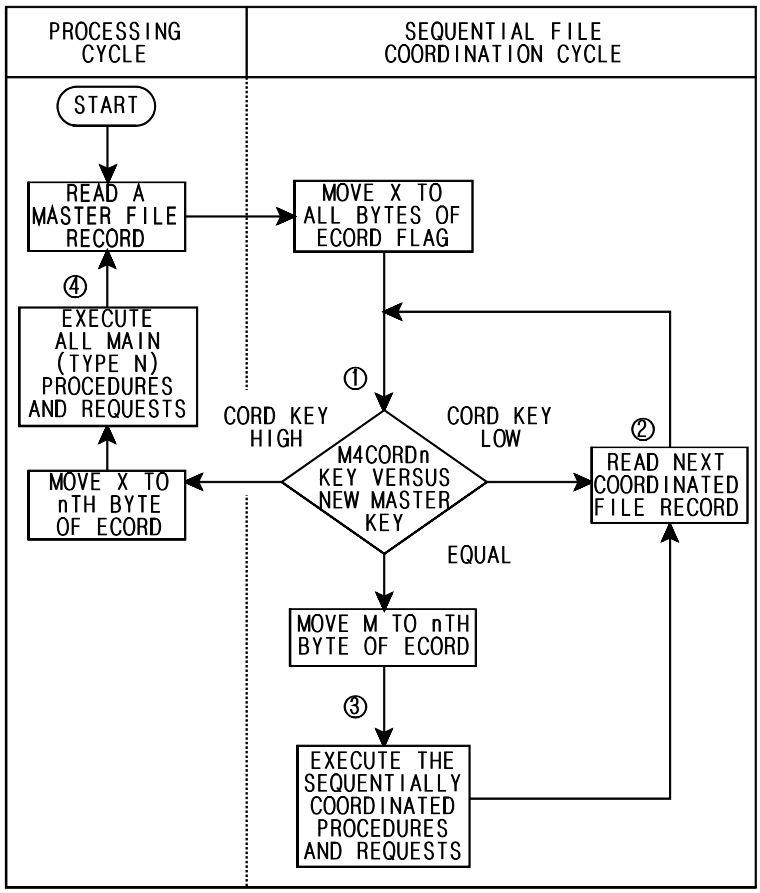



Figure 6-10 Flow of Single Coordinated File Match Only Processing

The following comments refer to the numbered areas in Figure 6-10.

**Note:** In Figure 6-10, the n in M4CORDn is the number of the only sequentially coordinated file in the run.

1. The key(s) of the M4CORDn file and the master file are compared. For file alignment purposes, VISION:Builder considers that any file that is at end of file has a key value of high values (all nines for a numeric key field, hex FF for a character key).
2. Note that non-matching coordinated file records are bypassed. You do not get the opportunity to access for processing or output any of the non-matching sequentially coordinated file records.
3. Sequentially coordinated procedures contain references (not necessarily executed) to a sequentially coordinated file field or to the ECORD flag. The nth byte of ECORD is an M when these procedures are executed at this point in the program logic. The master file work areas and the coordinated file (M4CORDn) work areas are available during execution of the sequentially coordinated procedures.
4. This is either the follow-up pass for a master record after all matching coordinated file records are processed or the only pass of the master record against the Main (Normal) procedures in the case of a master record with no matching coordinated file records. All Main (Normal) procedures (sequentially coordinated or not) are executed at this point in the program logic. Only the master file work areas are available to the procedures.

For the purposes of illustration, assume you have a sequential file of customer history records that needs to be processed with the application's customer file. This history file has non-structured records that have two key fields: customer number and order number. The customer file is keyed only by customer number at the root level. To process all of a customer's history file records with that customer's customer file record, assign the customer file as the master file and the history file as the sequentially coordinated file. If the history file were assigned as the master file, there might be a condition where more than one record with the same match key value on the master file (the history file) matched to a coordinated file record.

Coordination of the customer file occurs only with the first history file record for a given customer number. The files coordinate only on the customer number. This is because VISION:Builder automatically constructs the match key according to the file with the fewest number of file definition key fields. So, you do not need to specify the KEYNAME keyword on the FILE CORDn command. Figure 6-11 shows the entries on the FILE CORDn command needed to establish the sequential coordination relationship.

```
CONTROL NAME F150 TERM LISTGEN
FILE MASTER INPUT  CUSTOMER
FILE CORD1 CUSTHIST MATCHONLY
FILE REPORT
```

Figure 6-11 Single-File Sequential Coordination Match Only: Run Control Commands

**Note:** For a complete version of this example, see Single-File Sequential Coordination Match Only: The Run Control Commands – Part 1 of 2 on page D-35.

Note that in this application, you only need to access the history records that match a current customer record. The MATCHONLY keyword on the FILE CORDn command indicates that only matching records should be processed.

The table on page 6-23 and Figure 6-12 on page 6-24 illustrate the interaction of the sequential file coordination cycle and the processing cycle, the alignment of the files by keys, when records are available in their work areas, and what procedures can be executed during these cycles.

The table starting on page 6-23 lists the keys of the records in the customer file and the customer history file. The values of the keys in the two files are what drives the cycles.

Figure 6-12 on page 6-24 shows the number of times (cycle count reference number) that VISION:Builder will go through the flow in Figure 6-10 on page 6-21 to process and make available to the application's procedures the records from the two files shown in the following table.

**Note:** The comparison and alignment are done using the CUSTNO field only. Sequence checking is done using all key fields within each file.

| Customer Master File | Customer History File | |
|---|---|---|
| CUSTNO | CUSTNO | ORDNO |
| 00002 | 00001 | 10001 |
| 00003 | 00001 | 10004 |
| 00005 | 00001 | 10005 |
| 00010 | 00002 | 20003 |
| 00015 | 00002 | 20005 |
| | 00003 | 30001 |
| | 00010 | 01001 |
| | 00010 | 01004 |
| | 00020 | 02003 |

```
                                    SINGLE FILE SEQUENTIAL COORDINATION                                    PAGE 1
                                    LIST THE FILE KEYS AND SYSTEM FLAGS

        <MASTER>  <....CORD 1....>  <BUILDER FlAGS>

-------------------------------------------------------------------------------------------------------------------------
  CYCLE   CUSTOMER  CUSTOMER  ORDER   ECORD  MISSPASS        PROCESS CYCLE
  COUNT    NUMBER    NUMBER   NUMBER                          IN CONTROL
-------------------------------------------------------------------------------------------------------------------------
      1   00002     00002    20003    M                 SEQUENTIAL COORDINATION
      2   00002     00002    20005    M                 SEQUENTIAL COORDINATION
      3   00002        -        -     X       M         STANDARD PROCESSING
      4   00003     00003    30001    M                 SEQUENTIAL COORDINATION
      5   00003        -        -     X       M         STANDARD PROCESSING
      6   00005        -        -     X       X         STANDARD PROCESSING
      7   00010     00010    01001    M                 SEQUENTIAL COORDINATION
      8   00010     00010    01004    M                 SEQUENTIAL COORDINATION
      9   00010        -        -     X       M         STANDARD PROCESSING
     10   00015        -        -     X       X         STANDARD PROCESSING
```

Figure 6-12 Single File Sequential Coordination Listing the File Keys and System Flags

The keys of the two files are shown from their work areas: MASTER and CORD 1. The record and its data are available when a key value is shown. The dashes (--) indicate that the record and data are not available at that point within the flow through the cycles.

The value in the first byte of the ECORD flag is shown. It indicates the relationship of the customer history file record (M4CORD1) to the current customer record (the master file).

- The value M indicates a match has occurred between the master and coordinated file. You are in the sequential file coordination cycle ready to execute any sequentially coordinated Main (Normal) procedures. Both the current customer record (the master file) and the customer history record (M4CORD1) are available in their work areas.
- The X indicates that there are no more, or were not any, matches between the master and coordinated file. You are back in the standard processing cycle ready to execute all (sequentially coordinated or not) Main (Normal) procedures. This is the follow-up pass.

The text in the Process Cycle column indicates when sequentially coordinated and Main (Normal) procedures would be given control from VISION:Builder to perform their algorithms and/or outputs. The cycle in control at that point in the flow is indicated at the bottom of these columns.

The MISSPASS column shows the settings in the MISSPASS flag. This value can be interrogated during the follow-up pass in the processing cycle through all Main (Normal) procedures to determine whether or not matches occurred during the sequential file coordination cycle. A value of M means that at least one match occurred. A value of X means that no matches occurred. A blank value indicates that the follow-up pass is not in progress.

## All Record Coordinated File Processing

Figure 6-13 illustrates the flow for a master file and one sequentially coordinated file, when you want to process all records in the coordinated file, matched or not.

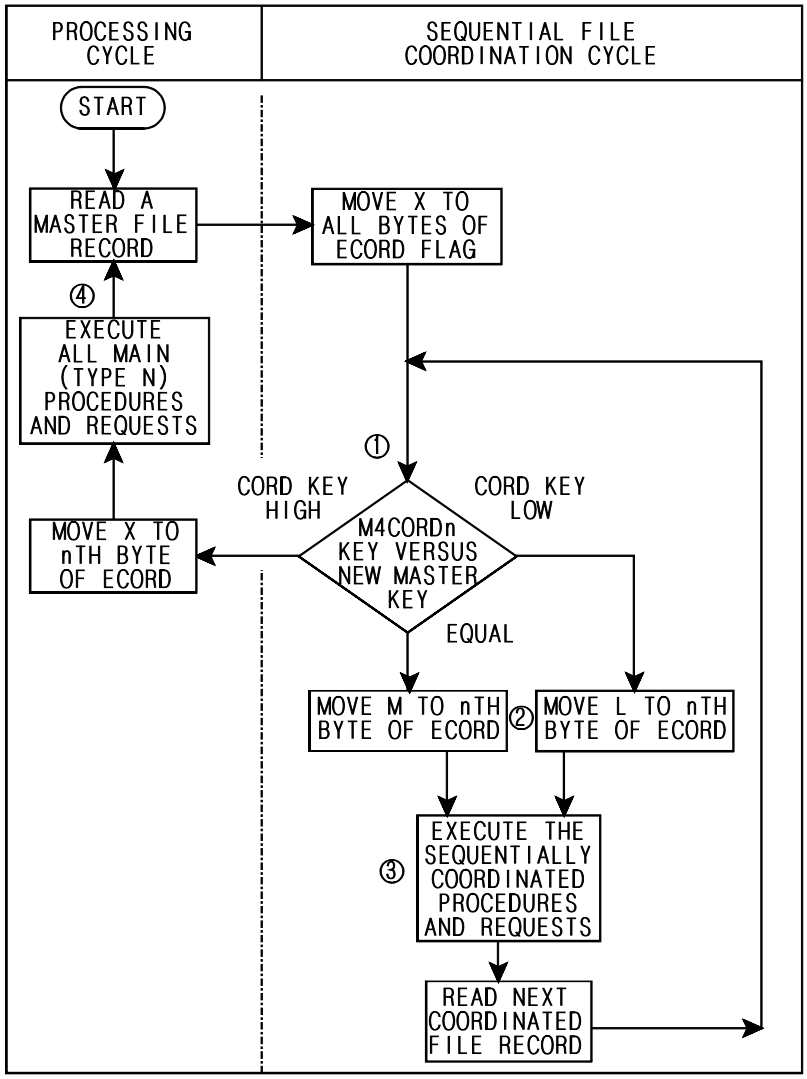



Figure 6-13 Flow of Single Coordinated File, All Record Processing

The following notes refer to the numbered areas in Figure 6-13.

**Note:** In Figure 6-13, the n in M4CORDn is the number of the only sequentially coordinated file in the run.

1 The key(s) of the M4CORDn file and the master file are compared. For file alignment purposes, VISION:Builder considers that any file that is at end of file has a key value of high values (all nines for a numeric key field, hex FF for a character key).

2 Note that the ECORD flag is set to reflect the alignment check. The value M indicates a match; L indicates that the sequentially coordinated file key is low when compared to the master file key. Because you are going to be processing all records in the sequentially coordinated file, the ECORD setting helps you determine which actions are taken in sequentially coordinated procedures.

3 Sequentially coordinated procedures contain references (not necessarily executed) to a sequentially coordinated file field or to the ECORD flag. At this point in the program logic, the nth byte of ECORD will be M or L when these procedures are executed. The master file work areas and the coordinated file (M4CORDn) work areas are available during execution of the sequentially coordinated procedures. When ECORD is M, the master file work area contains the record with a match key equal to that in the coordinated file. When ECORD is L, the master file work area contains the record with a match key value greater than that in the coordinated file.

4 This is either the follow-up pass for a master record after all matching and/or low cycles against this master file record have been processed or the only pass of the master record against the Main (Normal) procedures in the case of a master record with no low or matching coordinated file records. All Main (Normal) procedures (sequentially coordinated or not) are executed at this point in the program logic. Only the master file work area(s) are available to the procedures.

For example, assume you have a sequential file of customer order history records that needs to be processed with the application's customer file. This history file has non-structured records that have two key fields: customer number and order number. The customer file is keyed only by customer number at the root level. To process all of a customer's history file records with that customer's customer file record, assign the customer file as the master file and the history file as the sequentially coordinated file. If the history file was assigned as the master file, there might be a condition where more than one record with the same match key value on the master file (the history file) would match to a coordinated file record.

Coordination of the customer file occurs only with the first history file record for a given customer number. The files coordinate only on the customer number. This is because VISION:Builder automatically constructs the match key according to the file with the fewest number of file definition key fields. So, you do not need to specify the KEYNAME keyword on the FILE CORDn command. See Figure 6-14 for the entries on the FILE CORDn command needed to establish the sequential coordination relationship.

```
CONTROL NAME F152 TERM LISTGEN
FILE MASTER INPUT  CUSTOMER
FILE CORD1 CUSTHIST ALLRECS
FILE REPORT
```

Figure 6-14 Single-File Sequential Coordination All Records: Run Control Commands

**Note:** For a complete version of this example, see Single-File Sequential Coordination All Records: The Run Control Commands – Part 1 of 2 on page D-38.

In this application, you need to access all history records whether or not they match to current customer records. Therefore, you must specify the ALLRECS keyword on the FILE CORDn command to indicate that all record processing is wanted.

The table on pages 6-27 and Figure 6-15 on page 6-28 illustrate the interaction of the sequential file coordination cycle and the processing cycle, the alignment of the files by keys, when records are available in their work areas, and what procedures can be executed during these cycles.

The table starting on page 6-27 lists the keys of the records in the customer file and the customer-order history file. The values of the keys in the two files is what drives the cycles.

Figure 6-15 on page 6-28 shows the number of times (cycle count reference number) that VISION:Builder will go through the flow in Figure 6-13 to process and make available to the application's procedures the records from the two files shown in the following table.

**Note:** The comparison and alignment are done using the CUSTNO field only. Sequence checking is done using all key fields within each file.

| Customer Master File | Customer History File | |
|---|---|---|
| CUSTNO | CUSTNO | ORDNO |
| 00002 | 00001 | 10001 |
| 00003 | 00001 | 10004 |
| 00005 | 00001 | 10005 |
| 00010 | 00002 | 20003 |
| 00015 | 00002 | 20005 |
| | 00003 | 30001 |
| | 00010 | 01001 |
| | 00010 | 01004 |
| | 00020 | 02003 |

```
                                    ALL RECORD COORDINATED FILE PROCESSING                    PAGE 1
                                    LIST THE FILE KEYS AND SYSTEM FLAGS

          <MASTER>  <....CORD 1....>  <BUILDER FlAGS>

-------------------------------------------------------------------------------------------------------------
  CYCLE   CUSTOMER  CUSTOMER  ORDER   ECORD  MISSPASS        PROCESS CYCLE
  COUNT    NUMBER    NUMBER   NUMBER                          IN CONTROL
-------------------------------------------------------------------------------------------------------------
      1   00002     00001    10001     L            SEQUENTIAL COORDINATION
      2   00002     00001    10004     L            SEQUENTIAL COORDINATION
      3   00002     00001    10005     L            SEQUENTIAL COORDINATION
      4   00002     00002    20003     M            SEQUENTIAL COORDINATION
      5   00002     00002    20005     M            SEQUENTIAL COORDINATION
      6   00002       -        -       X       M    STANDARD PROCESSING
      7   00003     00003    30001     M            SEQUENTIAL COORDINATION
      8   00003       -        -       X       M    STANDARD PROCESSING
      9   00005       -        -       X       X    STANDARD PROCESSING
     10   00010     00010    01001     M            SEQUENTIAL COORDINATION
     11   00010     00010    01004     M            SEQUENTIAL COORDINATION
     12   00010       -        -       X       M    STANDARD PROCESSING
     13   00015       -        -       X       X    STANDARD PROCESSING
     14     -       00020    02003     L            SEQUENTIAL COORDINATION
     15     -         -        -       X       X    STANDARD PROCESSING
```

Figure 6-15 All Record Coordinated File Processing Listing the File Keys and System Flags

The keys of the two files are shown from their work areas: MASTER and CORD 1. The record and its data are available when a key value is shown. The dashes (--) indicate that the record and data are not available at that point. The EOF value indicates that the master is at end of file. For alignment purposes, VISION:Builder considers that any file at end of file has a key value of high values.

The value in the first byte of the ECORD flag is shown. It indicates the relationship of the customer-order history file record (M4CORD1) to the current customer record (the master file).

- The value M indicates a match has occurred between the master and coordinated file. You are in the sequential file coordination cycle ready to execute any sequentially coordinated Main (Normal) procedures. Both the current customer record (the master file) and the customer-order history record (M4CORD1) are available in their work areas.
- The value L indicates that the coordinated file record key is low when compared to the current master file record key. This indicates an unmatched sequentially coordinated file record is available in the work area. You are in the sequential file coordination cycle ready to execute any sequentially coordinated Main (Normal) procedures. Both the current customer record (the master file) and the customer-order history record (M4CORD1) are available in their work areas. However, the record keys (customer number) of the records available do not match.
- The X indicates that there are no more, or were not any, matches between the master and coordinated file. You are back in the processing cycle ready to execute all (sequentially coordinated or not) Main (Normal) procedures. This is the follow-up pass.

The text in the Process Cycle column indicates when sequentially coordinated and Main (Normal) procedures would be given control from VISION:Builder to perform their algorithms and/or outputs.

The MISSPASS column shows the settings in the MISSPASS flag. This value can be interrogated during the follow-up pass in the processing cycle through all Main (Normal) procedures to determine whether or not matches occurred during the sequential file coordination cycle. A value of M means at least one match occurred. A value of X means no matches occurred. A blank value indicates that the follow-up pass is not in progress.

## Multiple File Sequential Coordination

The concepts for multiple file coordination build upon those of single file coordination. With more than one sequentially coordinated file, you must choose one of two basic processing methods.

- All sequentially coordinated files will match to the master file.
- One sequentially coordinated file matches to the master file, while all the other sequentially coordinated files match to other sequentially coordinated files. This is called chained multi-file sequential coordination. This is described later in this chapter.

With all sequentially coordinated files matching to the master file, you have the choice of nested or parallel processing of the sequentially coordinated file records. This choice dictates the order in which VISION:Builder advances to the next record in the respective sequentially coordinated files.

- Nested processing. VISION:Builder first reads the highest numbered sequentially coordinated file's records until that file's record key value is higher than the current master record, the next highest numbered sequentially coordinated file's records are read, and so on. The sequentially coordinated procedures are executed each time a sequentially coordinated file is advanced to its next matching record. With nested processing, you only have sequentially coordinated file records that match to the master file available for processing.
- Parallel processing. VISION:Builder reads all the sequentially coordinated files in parallel (at the same time), stopping an individual file's reads only when its match key value is higher than the current master file record's match key. The sequentially coordinated Main (Normal) procedures are executed each time sequentially coordinated files are advanced to their next low or matching records. With parallel processing, specify either matching record only or all record processing.

A very important point to remember when using multi-file sequential coordination is that when there are a different number of key fields between the matching file and the coordinating files, the file with the fewest number of keys determines the number of key fields used when comparing and aligning the files. However, sequence checking on each file is still performed using all the key fields defined for the file.

### Nested Multi-File Sequential Coordination

Figure 6-16 illustrates the flow of processing when multiple sequentially coordinated files are used in a VISION:Builder application. These figures use two sequentially coordinated files, M4CORD1 and M4CORD2. Once reviewed, it should be easy to extend these charts to include more than two sequentially coordinated files.

Figure 6-16 shows the flow for processing multiple sequentially coordinated files using nested processing.

The following comments relate to Figure 6-16.

- With nested processing, the sequential file coordination cycle initially aligns the sequentially coordinated file records with the master file records and all the sequentially coordinated files so that each file is positioned to a record with a key value that is equal to or higher than the key value of the current master file record. All sequentially coordinated file records with key values lower than the key value of the current master file record are bypassed. Nested processing only makes the sequentially coordinated file records that match master file records available for processing.
- After the initial alignment of the sequentially coordinated file records with the current master file record, if at least one sequentially coordinated file has a record that matches to the current master file record, all sequentially coordinated Main (Normal) procedures are given control for execution. If no matches occur, control is immediately returned to the normal processing cycle. This is the follow-up pass.
- After the initial alignment steps, the process continues using a different set of steps. With nested processing, alignment checks and the reading of records are performed against the sequentially coordinated files in descending order based on their assigned number (M4CORD9 through M4CORD1) in the application. If a sequentially coordinated file has a matching record in the previous alignment process, another record is read and the keys are compared to the current master record keys. If a match occurs, the ECORD flag is set and control is given to the sequentially coordinated Main (Normal) procedures. Note that once a match occurs, alignment and reading of the other sequentially coordinated files are postponed until later. All matching records from the highest numbered sequentially coordinated file are processed before the next highest file is checked; this is the nesting process.
- Once all matching records from all sequentially coordinated files are processed, control returns to the normal processing cycle and all Main (Normal) procedures are given control for execution. This is the follow-up pass.

## Parallel Multi-File Sequential Coordination

Figure 6-16 is the flow for processing multiple sequentially coordinated files using parallel processing.

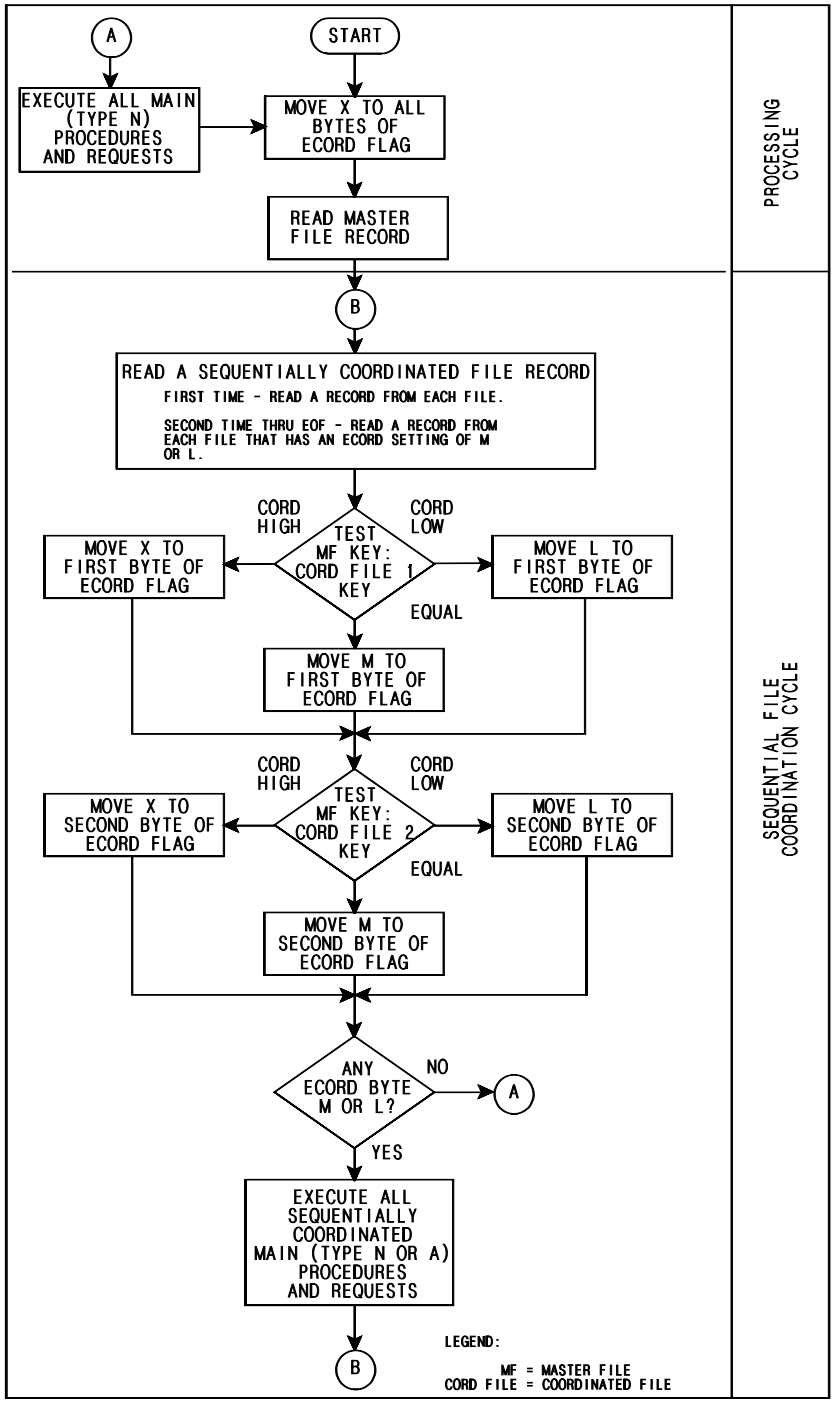



Figure 6-16 Processing Flow for Parallel Multi-File Sequential Coordination

The following comments relate to Figure 6-16.

- With parallel processing, the sequential file coordination cycle has only one place in the flow where the alignment of the sequentially coordinated file records with the master file records occurs. The very first time the process is entered, a record from each sequentially coordinated file is read. This is done to get the whole process started. The second and all subsequent times through the flow, records are read only from a sequentially coordinated file that has a M or L setting in its ECORD flag byte.
- After the appropriate reads are performed, the keys from each and every sequentially coordinated file record are compared to the keys of the current master file record. The ECORD flag byte for the sequentially coordinated file is set to M, L, or X (for equal, low, or high, respectively) to reflect its relationship with the current master file record.
- Once all sequentially coordinated file records have been tested against the current master file record, if any sequentially coordinated file had a record that matched or was low (ECORD M or L) when compared to the current master file record, all sequentially coordinated Main (Normal) procedures are given control for execution.
- Once all matching and low records from all sequentially coordinated files are processed, control returns to the normal processing cycle and all Main (Normal) procedures are given control for execution. This is the follow-up pass.

The following examples and tables illustrate the multiple file sequential coordination process. What is shown is the interaction of the sequential file coordination cycle and the processing cycle, the alignment of the files by keys, when records are available in their work areas, and what procedures can be executed during these cycles.

The following table shows the key values of the records in the three files used in the following examples. The master file is the customer file in sequence by customer number. For the sequentially coordinated input files, M4CORD1 is used as a customer order history file sequenced by customer number and order number. The M4CORD2 file is an item history file sequenced by customer, order, and item number.

**Note:** The comparison and alignment are done using the CUSTNO field only. This is because both sequentially coordinated files are matching to the master file that only has one key field, CUSTNO. Sequence checking is done using all fields within each file.

| Customer Master File | Customer History File | | Item History File | | |
|---|---|---|---|---|---|
| CUSTNO | CUSTNO | ORDNO | CUSTNO | ORDNO | ITEMNO |
| 00002 | 00001 | 10001 | 00001 | 10001 | M55 |
| 00003 | 00001 | 10004 | 00001 | 10001 | N94 |
| 00005 | 00001 | 10005 | 00001 | 10004 | M52 |
| 00010 | 00002 | 20003 | 00001 | 10004 | O17 |
| 00015 | 00002 | 20005 | 00001 | 10005 | O13 |
| | 00003 | 30001 | 00002 | 20003 | H11 |
| | 00010 | 01001 | 00002 | 20003 | H68 |
| | 00010 | 01004 | 00002 | 20009 | E68 |
| | 00020 | 02003 | 00010 | 01001 | N34 |
| | | | 00010 | 01004 | E68 |
| | | | 00010 | 01004 | M52 |
| | | | 00015 | 15001 | O17 |
| | | | 00015 | 15006 | N11 |
| | | | 00015 | 15006 | N19 |
| | | | 00020 | 02003 | E13 |

Figure 6-17 and Figure 6-18 on page 6-34 show how to specify nested processing and how the files and records will be processed.

On the FILE CORDn commands (RF statement) in Figure 6-17, the MATCHONLY keywords for both M4CORD1 and M4CORD2 indicate to VISION:Builder that both files will match to the master file, only matched sequentially coordinated file records are needed for processing, and nested processing is performed.

```
CONTROL NAME F155 TERM LISTGEN
FILE MASTER INPUT  CUSTOMER
FILE CORD1 CUSTHIST MATCHONLY
FILE CORD2 ITEMHIST MATCHONLY
FILE REPORT
```

Figure 6-17 Nested Multi-File Sequential Coordination: Run Control Commands

**Note:** For a complete version of this example, see Nested Multi-File Sequential Coordination: Run Control Commands – Part 1 of 2 on page D-41.

Figure 6-18 shows the number of times (cycle count reference number) that VISION:Builder will go through the flow in Figure 6-16 on page 6-31 to process and make available to the application's procedures the records from the three files shown in the table starting on page 6-33.

NESTED MULTI-FILE SEQUENTIAL COORDINATION PAGE 1
LIST THE FILE KEYS AND SYSTEM FLAGS

| | <MASTER> | <....CORD 1....> | | <.....CORD 2.....> | | | <BUILDER FLAGS> | | |
|---|---|---|---|---|---|---|---|---|---|
| CYCLE COUNT | CUSTOMER NUMBER | CUSTOMER NUMBER | ORDER NUMBER | CUSTOMER NUMBER | ORDER NUMBER | ITEM NUMBER | ECORD | MISSPASS | PROCESS CYCLE IN CONTROL |
| 1 | 00002 | 00002 | 20003 | 00002 | 20003 | H11 | MM | | SEQUENTIAL COORDINATION |
| 2 | 00002 | 00002 | 20003 | 00002 | 20003 | H68 | MM | | SEQUENTIAL COORDINATION |
| 3 | 00002 | 00002 | 20003 | 00002 | 20009 | E68 | MM | | SEQUENTIAL COORDINATION |
| 4 | 00002 | 00002 | 20005 | - | - | - | MX | | SEQUENTIAL COORDINATION |
| 5 | 00002 | - | - | - | - | - | XX | M | STANDARD PROCESSING |
| 6 | 00003 | 00003 | 30001 | 00003 | 30001 | E68 | MM | | SEQUENTIAL COORDINATION |
| 7 | 00003 | - | - | - | - | - | XX | M | STANDARD PROCESSING |
| 8 | 00005 | - | - | - | - | - | XX | X | STANDARD PROCESSING |
| 9 | 00010 | 00010 | 01001 | 00010 | 01001 | N34 | MM | | SEQUENTIAL COORDINATION |
| 10 | 00010 | 00010 | 01001 | 00010 | 01004 | E68 | MM | | SEQUENTIAL COORDINATION |
| 11 | 00010 | 00010 | 01001 | 00010 | 01004 | M52 | MM | | SEQUENTIAL COORDINATION |
| 12 | 00010 | 00010 | 01004 | - | - | - | MX | | SEQUENTIAL COORDINATION |
| 13 | 00010 | - | - | - | - | - | XX | M | STANDARD PROCESSING |
| 14 | 00015 | - | - | 00015 | 15001 | O17 | XM | | SEQUENTIAL COORDINATION |
| 15 | 00015 | - | - | 00015 | 15006 | N11 | XM | | SEQUENTIAL COORDINATION |
| 16 | 00015 | - | - | 00015 | 15006 | N19 | XM | | SEQUENTIAL COORDINATION |
| 17 | 00015 | - | - | - | - | - | XX | M | STANDARD PROCESSING |

Figure 6-18 Nested Multi-File Sequential Coordination Listing the File Keys and System Flags

The keys of the three files are shown from their work areas: MASTER, CORD 1, and CORD 2. The record and its data are available when a key value is shown. The dashes (--) indicate that the record and data are not available at that point.

The values in byte one and two of the ECORD flag are shown. The value indicates the relationship of the customer order history file record (M4CORD1) and the item history file record (M4CORD2) to the current customer record (the master file).

- The value M indicates a match between the master and a coordinated file. You are in the sequential file coordination cycle ready to execute any sequentially coordinated Main (Normal) procedures. The current customer record (the master file) along with a matching customer order history record (M4CORD1) and/or a matching item history record (M4CORD2) are available in their work areas.
- The X in one or the other of the ECORD flag bytes indicates that the particular record (M4CORD1 or M4CORD2) is not available for processing at that point within the sequential file coordination cycle.
- The XX indicates that there are no more, or were not any, matches between the master and the two sequentially coordinated files. You are back in the processing cycle ready to execute all (sequentially coordinated or not) Main (Normal) procedures. This is the follow-up pass.

The text in the Process Cycle column indicates when sequentially coordinated and Main (Normal) procedures would be given control from VISION:Builder to perform their algorithms and/or outputs to satisfy the application's requirements.

The MISSPASS column shows the settings in the MISSPASS flag. This value can be interrogated during the follow-up pass in the processing cycle through all Main (Normal) procedures to determine whether or not matches occurred during the sequential file coordination cycle. A value of M means at least one match occurred. A value of X means no matches occurred. A blank value indicates that the follow-up pass is not in progress.

Figure 6-19 and Figure 6-20 on page 6-36 show how to specify parallel processing and how the files and records are processed. By using parallel processing instead of nested, a different set of events and cycles occurs. You set up the application to process the three input files just like the example shown in Figure 6-17 on page 6-33 and Figure 6-18 on page 6-34, except you specify parallel processing.

On the FILE CORDn command in Figure 6-19 (RF statement), you specify the CHAIN TO M keyword and value for both M4CORD1 and M4CORD2, which indicates the following to VISION:Builder:

- Both files match to the master file.
- Only matched sequentially coordinated file records are needed for processing.
- Parallel processing is performed.

```
CONTROL NAME F156 TERM LISTGEN
FILE MASTER INPUT  CUSTOMER
FILE CORD1 CUSTHIST CHAIN TO M
FILE CORD2 ITEMHIST CHAIN TO M
FILE REPORT
```

Figure 6-19 Parallel Multi-File Sequential Coordination Chained to the Master File: Run Control Commands

**Note:** For a complete version of this example, see Parallel Multi-File Sequential Coordination Chained to the Master File: Run Control Commands – Part 1 of 2 on page D-44.

Figure 6-20 shows the number of times (cycle count reference number) that VISION:Builder will go through the flow in Figure 6-16 on page 6-31 to process and make available to the application's procedures the records from the three files shown in the table starting on page 6-33.

PARALLEL MULTI-FILE SEQUENTIAL COORDINATION PAGE 1
LIST THE FILE KEYS AND SYSTEM FLAGS

<MASTER> <....CORD 1....> <.....CORD 2.....> <BUILDER FLAGS>

| CYCLE COUNT | CUSTOMER NUMBER | CUSTOMER NUMBER | ORDER NUMBER | CUSTOMER NUMBER | ORDER NUMBER | ITEM NUMBER | ECORD | MISSPASS | PROCESS CYCLE IN CONTROL |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 00002 | 00002 | 20003 | 00002 | 20003 | H11 | MM | | SEQUENTIAL COORDINATION |
| 2 | 00002 | 00002 | 20005 | 00002 | 20003 | H68 | MM | | SEQUENTIAL COORDINATION |
| 3 | 00002 | - | - | 00002 | 20009 | E68 | XM | | SEQUENTIAL COORDINATION |
| 4 | 00002 | - | - | - | - | - | XX | M | STANDARD PROCESSING |
| 5 | 00003 | 00003 | 30001 | 00003 | 30001 | E68 | MM | | SEQUENTIAL COORDINATION |
| 6 | 00003 | - | - | - | - | - | XX | M | STANDARD PROCESSING |
| 7 | 00005 | - | - | - | - | - | XX | X | STANDARD PROCESSING |
| 8 | 00010 | 00010 | 01001 | 00010 | 01001 | N34 | MM | | SEQUENTIAL COORDINATION |
| 9 | 00010 | 00010 | 01004 | 00010 | 01004 | E68 | MM | | SEQUENTIAL COORDINATION |
| 10 | 00010 | - | - | 00010 | 01004 | M52 | XM | | SEQUENTIAL COORDINATION |
| 11 | 00010 | - | - | - | - | - | XX | M | STANDARD PROCESSING |
| 12 | 00015 | - | - | 00015 | 15001 | O17 | XM | | SEQUENTIAL COORDINATION |
| 13 | 00015 | - | - | 00015 | 15006 | N11 | XM | | SEQUENTIAL COORDINATION |
| 14 | 00015 | - | - | 00015 | 15006 | N19 | XM | | SEQUENTIAL COORDINATION |
| 15 | 00015 | - | - | - | - | - | XX | M | STANDARD PROCESSING |

Figure 6-20 Parallel Multi-File Sequential Coordination Listing the File Keys and System Flags

The keys of the three files are shown in their work areas. The record and its data are available when a key value is shown. The dashes (--) indicate that the record and data are not available at that point.

The values in byte 1 and 2 of the ECORD flag are shown. The value indicates the relationship of the customer order history file record (M4CORD1) and the item history file record (M4CORD2) to the current customer record (the master file).

- The value M indicates a match between the master and a coordinated file. You are in the sequential file coordination cycle ready to execute any sequentially coordinated Main (Normal) procedures. The current customer record (the master file) along with a matching customer order history record (M4CORD1) and/or a matching item history record (M4CORD2) are available in their work areas.
- The X in one or the other of the ECORD flag bytes indicates that the particular record (M4CORD1 or M4CORD2) is not available for processing at that point within the sequential file coordination cycle.
- The XX indicates that there are no more, or were not any, matches between the master and the two sequentially coordinated files. You are back in the processing cycle ready to execute all (sequentially coordinated or not) Main (Normal) procedures. This is the follow-up pass.

The text in the Process Cycle column indicates when sequentially coordinated and Main (Normal) procedures would be given control from VISION:Builder to perform their algorithms and/or outputs.

The MISSPASS column shows the settings in the MISSPASS flag. This value can be interrogated during the follow-up pass in the processing cycle through all Main (Normal) procedures to determine whether or not matches occurred during the sequential file coordination cycle. A value of M means at least one match occurred. A value of X means no matches occurred. A blank value indicates that the follow-up pass is not in progress.

Figure 6-21 and Figure 6-22 show how to specify parallel processing and how the files and records are processed. In this example, you set up the application to process the three input files just like the example shown in Figure 6-19 and Figure 6-20, except you specify that you want all record processing (that is, access to unmatched sequentially coordinated file records).

On the FILE CORDn command in Figure 6-21 (RF statement), specify the ALLRECS keyword for both M4CORD1 and M4CORD2 to indicate that both files are matched to the master file, all sequentially coordinated file records (matched and unmatched) are needed for processing.

```
CONTROL NAME F157 TERM LISTGEN
FILE MASTER INPUT  CUSTOMER
FILE CORD1 CUSTHIST ALLRECS
FILE CORD2 ITEMHIST ALLRECS
FILE REPORT
```

Figure 6-21 Parallel Multi-File Sequential Coordination All Records Processing Matching to the Master File: Run Control Commands

**Note:** For a complete version of this example, see Parallel Multi-File Sequential Coordination All Records Processing Matching to the Master File: Run Control Commands – Part 1 of 2 on page D-47.

Figure 6-22 shows the number of times (cycle count reference number) that VISION:Builder will go through the flow in Figure 6-16 on page 6-31 to process and make available to the application's procedures the records from the three files shown in the table starting on page 6-33. Sequential coordination:

PARALLEL MULTI-FILE SEQUENTIAL COORDINATION - ALL RECORDS PAGE 1
LIST THE FILE KEYS AND SYSTEM FLAGS

| | <MASTER> | <....CORD 1....> | | <.....CORD 2.....> | | | <BUILDER FLAGS> | | |
|---|---|---|---|---|---|---|---|---|---|
| CYCLE COUNT | CUSTOMER NUMBER | CUSTOMER NUMBER | ORDER NUMBER | CUSTOMER NUMBER | ORDER NUMBER | ITEM NUMBER | ECORD | MISSPASS | PROCESS CYCLE IN CONTROL |
| 1 | 00002 | 00001 | 10001 | 00001 | 10001 | M55 | LL | | SEQUENTIAL COORDINATION |
| 2 | 00002 | - | - | 00001 | 10001 | N94 | XL | | SEQUENTIAL COORDINATION |
| 3 | 00002 | 00001 | 10004 | 00001 | 10004 | M52 | LL | | SEQUENTIAL COORDINATION |
| 4 | 00002 | - | - | 00001 | 10004 | O17 | XL | | SEQUENTIAL COORDINATION |
| 5 | 00002 | 00001 | 10005 | 00001 | 10005 | O13 | LL | | SEQUENTIAL COORDINATION |
| 6 | 00002 | 00002 | 20003 | 00002 | 20003 | H11 | MM | | SEQUENTIAL COORDINATION |
| 7 | 00002 | 00002 | 20005 | 00002 | 20003 | H68 | MM | | SEQUENTIAL COORDINATION |
| 8 | 00002 | - | - | 00002 | 20009 | E68 | XM | | SEQUENTIAL COORDINATION |
| 9 | 00002 | - | - | - | - | - | XX | M | STANDARD PROCESSING |
| 10 | 00003 | 00003 | 30001 | 00003 | 30001 | E68 | MM | | SEQUENTIAL COORDINATION |
| 11 | 00003 | - | - | - | - | - | XX | M | STANDARD PROCESSING |
| 12 | 00005 | - | - | - | - | - | XX | X | STANDARD PROCESSING |
| 13 | 00010 | 00010 | 01001 | 00010 | 01001 | N34 | MM | | SEQUENTIAL COORDINATION |
| 14 | 00010 | 00010 | 01004 | 00010 | 01004 | E68 | MM | | SEQUENTIAL COORDINATION |
| 15 | 00010 | - | - | 00010 | 01004 | M52 | XM | | SEQUENTIAL COORDINATION |
| 16 | 00010 | - | - | - | - | - | XX | M | STANDARD PROCESSING |
| 17 | 00015 | - | - | 00015 | 15001 | O17 | XM | | SEQUENTIAL COORDINATION |
| 18 | 00015 | - | - | 00015 | 15006 | N11 | XM | | SEQUENTIAL COORDINATION |
| 19 | 00015 | - | - | 00015 | 15006 | N19 | XM | | SEQUENTIAL COORDINATION |
| 20 | 00015 | - | - | - | - | - | XX | M | STANDARD PROCESSING |
| 21 | - | 00020 | 02003 | 00020 | 02003 | E13 | LL | | SEQUENTIAL COORDINATION |
| 22 | - | - | - | - | - | - | XX | X | STANDARD PROCESSING |

Figure 6-22 Parallel Multi-File Sequential Coordination - All Records

The keys of the three files are shown in their work areas. The record and its data are available when a key value is shown. The dashes (--) indicate that the record and data are not available at that point.

The values in byte 1 and 2 of the ECORD flag are shown. The value indicates the relationship of the customer order history file record (M4CORD1) and the item history file record (M4CORD2) to the current customer record (the master file).

- The value M indicates a match between the master and a coordinated file. You are in the sequential file coordination cycle ready to execute any sequentially coordinated Main (Normal) procedures. The current customer record (the master file) along with a matching customer order history record (M4CORD1) and/or a matching item history record (M4CORD2) are available in their work areas.
- The value L in one and/or the other ECORD flag bytes indicates that the particular sequentially coordinated file record key (M4CORD1 and/or M4CORD2) is low when compared to the current master file record key. This indicates an unmatched sequentially coordinated file record is available in the work area. You are in the sequential file coordination cycle ready to execute any sequentially coordinated Main (Normal) procedures. The current customer record (the master file) along with a low customer order history record (M4CORD1) and/or a low item history record are available in their work areas.

However, the record keys (customer number) of the sequentially coordinated file records that are available do not match the record key of the available master file record.

- The X in one or the other of the ECORD flag bytes indicates that the particular record (M4CORD1 or M4CORD2) is not available for processing at that point.
- The XX indicates that there are no more, or were not any, matches between the master and the two sequentially coordinated files. You are back in the processing cycle ready to execute all (sequentially coordinated or not) Main (Normal) procedures. This is the follow-up pass.

The text in the Process Cycle column indicates when sequentially coordinated and Main (Normal) procedures would be given control from VISION:Builder to perform their algorithms and/or outputs.

The MISSPASS column shows the settings in the MISSPASS flag. This value can be interrogated during the follow-up pass in the processing cycle through all Main (Normal) procedures to determine whether or not matches occurred during the sequential file coordination cycle. A value of M means at least one match occurred. A value of X means no matches occurred. A blank value indicates that the follow-up pass is not in progress.

## Chained Multi-File Sequential Coordination

With chained multi-file sequential file coordination, the sequentially coordinated files can match to the master or other sequentially coordinated files. However, one sequentially coordinated file must match to the master file.

A chain of sequentially coordinated files can be set up, for example, as shown in Figure 6-23. The M4CORD1 file is matched against the master file, while the M4CORD2 and M4CORD3 files are, in turn, matched against M4CORD1. The chain must start with the first sequentially coordinated file in the chain matching to the master file. The chain continues with the second sequentially coordinated file matching to the first, a third to the first, and so on. At the end of the chain, multiple sequentially coordinated files can match to the same file in the chain as long as they do not have files matching to them.

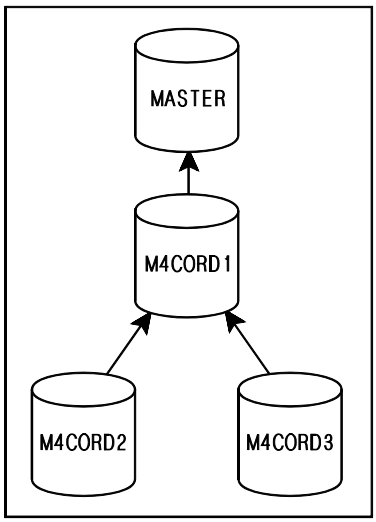



Figure 6-23 Diagram of Chained Multi-File Sequential Coordination

The sequentially coordinated file that matches to the master file does so in match only mode. You specify the FILE CORDn command with the CHAIN TO M keyword and value (or use the RF additional files statement) to indicate the match to the master file. The sequentially coordinated files that match to (chain to) another sequentially coordinated file are also processed in match only mode. Enter the number of the sequentially coordinated file that is to serve as the match file by using the the CHAIN TO keyword on the FILE CORDn command or the RF additional files statement.

Figure 6-24 shows the Run Control commands needed to specify the chained multi-file sequential coordination setup shown in Figure 6-23. Note the FILE CORD1 has CHAIN TO M, while FILE CORD2 and FILE CORD3 have 1 specified in the CHAIN TO keyword.

```
CONTROL NAME F159 TERM LISTGEN
FILE MASTER INPUT  CUSTOMER
FILE CORD1 CUSTHIST CHAIN TO M
FILE CORD2 CUSTHIST CHAIN TO 1
FILE CORD3 ITEMHIST CHAIN TO 1
FILE REPORT
```

Figure 6-24 Chained Multi-File Sequential Coordination: Run Control Commands

**Note:** For a complete version of this example, see Chained Multi-File Sequential Coordination: Run Control Commands – Part 1 of 2 on page D-50.

The method used to process these chained files is a combination of nested and parallel processing. The sequentially coordinated files are read and positioned so their coordinating keys are either equal to or greater than their match file. After each alignment check, the sequentially coordinated Main (Normal) procedures are given control. The ECORD flag settings indicate which records are available (matched). The process continues until all the sequentially coordinated files are positioned with the coordinating keys greater than their match file's keys.

After the initial alignment of all the sequentially coordinated files, the chained multi-file sequential coordination process starts at the end of the chain. The last sequentially coordinated file in the chain is processed in parallel against its match file. After each alignment check, the sequentially coordinated Main (Normal) procedures are given control. The ECORD flag settings indicate which records are available (matched).

The last sequentially coordinated file is processed until it is positioned at records with key values greater than its match file's key values; the ECORD flag setting is X. When the last sequentially coordinated file in the chain has its ECORD flag setting at X (a miss condition), the sequentially coordinated Main (Normal) procedures are given control one more time. This is a chained coordination miss pass.

The next sequentially coordinated file upward in the chain is aligned again by reading the next record in the file. This can cause more alignments with the sequentially coordinated files that are downward in the chain. The point to remember is that a match (upward) file is not aligned to its next record until its coordinating (downward) files have been through the chained coordination miss pass condition.

As the process moves upward through the chain of files, nested processing is used to align each sequentially coordinated file upward in the chain. Again, a match (upward) file will not align to its next record until its coordinating (downward) files have been through the chained coordination miss pass condition.

When all the chained sequentially coordinated files have been through the chained coordination miss pass condition, all the sequentially coordinated Main (Normal) procedures are given control one more time for the master file follow-up pass. The ECORD flag is set to Xs for all sequentially coordinated files and the MISSPASS flag is set to M or X.

The chained multi-file sequential coordination process requires that you use the ECORD flag and possibly the MISSPASS flag to determine where you are within the cycles that handle the processing.

The following example illustrates how a set of files and records are processed using chained multi-file sequential coordination. The example will use the file setup as shown in Figure 6-23 and Figure 6-24. The customer master file is in sequence by customer number. The two customer order history files (M4CORD1 and M4CORD2) are in sequence by customer number and order number. Finally, the item history file (M4CORD3) is in sequence by customer, order, and item number.

The following table shows the key values of the records in the four files you are using in this example. The master file and M4CORD1 are aligned by customer number. M4CORD2 and M4CORD3 are aligned with M4CORD1 by customer and order number.

| MASTER | M4CORD1 | | M4CORD2 | | M4CORD3 | | |
|---|---|---|---|---|---|---|---|
| Customer Master File | Customer History File 1 | | Customer History File 2 | | Item History File | | |
| CUSTNO | CUSTNO | ORDNO | CUSTNO | ORDNO | CUSTNO | ORDNO | ITEMNO |
| 00002 | 00001 | 10001 | 00001 | 10001 | 00001 | 10001 | M55 |
| 00003 | 00001 | 10004 | 00001 | 10004 | 00001 | 10001 | N94 |
| 00005 | 00001 | 10005 | 00001 | 10005 | 00001 | 10004 | M52 |
| 00010 | 00002 | 20003 | 00002 | 20003 | 00001 | 10004 | O17 |
| 00015 | 00002 | 20005 | 00002 | 20005 | 00001 | 10005 | O13 |
| | 00003 | 30001 | 00003 | 30001 | 00002 | 20003 | H11 |
| | 00010 | 01001 | 00010 | 01001 | 00002 | 20003 | H68 |
| | 00010 | 01004 | 00010 | 01004 | 00002 | 20009 | E68 |
| | 00020 | 02003 | 00020 | 02003 | 00010 | 01001 | N34 |
| | | | | | 00010 | 01004 | E68 |
| | | | | | 00010 | 01004 | M52 |
| | | | | | 00015 | 15001 | O17 |
| | | | | | 00015 | 15006 | N11 |
| | | | | | 00015 | 15006 | N19 |
| | | | | | 00020 | 02003 | E13 |

Figure 6-25 illustrates the chained multi-file sequential coordination process: the interaction of the sequential file coordination cycle and the processing cycle, the alignment of the files by keys, when records are available in their work areas, and what procedures can be executed during these cycles.

CHAINED MULTI-FILE SEQUENTIAL COORDINATION PAGE 1
USING BOTH NESTED AND PARALLEL COORDINATION
LIST THE FILE KEYS AND SYSTEM FLAGS

<MASTER> <....CORD 1....> <....CORD 2....> <.....CORD 3.....> <BUILDER FLAGS>

| CYCLE COUNT | CUSTOMER NUMBER | CUSTOMER NUMBER | ORDER NUMBER | CUSTOMER NUMBER | ORDER NUMBER | CUSTOMER NUMBER | ORDER NUMBER | ITEM NUMBER | ECORD | MISSPASS | PROCESS CYCLE IN CONTROL |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 00002 | 00002 | 20003 | 00002 | 20003 | 00002 | 20003 | H11 | MMM | | SEQUENTIAL COORDINATION |
| 2 | 00002 | 00002 | 20003 | - | - | 00002 | 20003 | H68 | MXM | | SEQUENTIAL COORDINATION |
| 3 | 00002 | 00002 | 20003 | - | - | - | - | - | MXX | | SEQUENTIAL COORDINATION |
| 4 | 00002 | 00002 | 20005 | 00002 | 20005 | - | - | - | MMX | | SEQUENTIAL COORDINATION |
| 5 | 00002 | 00002 | 20005 | - | - | - | - | - | MXX | | SEQUENTIAL COORDINATION |
| 6 | 00002 | - | - | - | - | - | - | - | XXX | M | STANDARD PROCESSING |
| 7 | 00003 | 00003 | 30001 | 00003 | 30001 | 00003 | 30001 | E68 | MMM | | SEQUENTIAL COORDINATION |
| 8 | 00003 | 00003 | 30001 | - | - | - | - | - | MXX | | SEQUENTIAL COORDINATION |
| 9 | 00003 | - | - | - | - | - | - | - | XXX | M | STANDARD PROCESSING |
| 10 | 00005 | - | - | - | - | - | - | - | XXX | X | STANDARD PROCESSING |
| 11 | 00010 | 00010 | 01001 | 00010 | 01001 | 00010 | 01001 | N34 | MMM | | SEQUENTIAL COORDINATION |
| 12 | 00010 | 00010 | 01001 | - | - | - | - | - | MXX | | SEQUENTIAL COORDINATION |
| 13 | 00010 | 00010 | 01004 | 00010 | 01004 | 00010 | 01004 | E68 | MMM | | SEQUENTIAL COORDINATION |
| 14 | 00010 | 00010 | 01004 | - | - | 00010 | 01004 | M52 | MXM | | SEQUENTIAL COORDINATION |
| 15 | 00010 | 00010 | 01004 | - | - | - | - | - | MXX | | SEQUENTIAL COORDINATION |
| 16 | 00010 | - | - | - | - | - | - | - | XXX | M | STANDARD PROCESSING |
| 17 | 00015 | - | - | - | - | - | - | - | XXX | X | STANDARD PROCESSING |

Figure 6-25 Chained Multi-File Sequential Coordination Using Both Nested and Parallel Coordination

The cycle count reference number indicates the number of times that VISION:Builder will go through the flow of the coordination process to make the records from the four files shown in the table starting on page 6-42 available to the application's procedures.

The keys of the four files are shown in their work areas. The record and its data are available when a key value is shown. The dashes (--) indicate that the record and data are not available at that point.

The values in bytes 1, 2, and 3 of the ECORD flag are shown. Byte 1 indicates the relationship of the customer order history file record (M4CORD1) and the customer master file record. Bytes 2 and 3 indicate the relationship of the other customer order history file (M4CORD2) and the item history file record (M4CORD3) to the current record in the M4CORD1 customer order history file.

- The value M indicates a match between the match file and its coordinating file(s). You will be in the sequential file coordination cycle ready to execute any sequentially coordinated Main (Normal) procedures. The current match file record along with its coordinating file(s) record(s) are available in their work areas.
- The value X indicates that the particular coordinating file's record is not available for processing at that point within the sequential file coordination cycle; there is no matching record at this time.
- The XXX indicates that there are no more, or were not any, matches between the master and the chain sequentially coordinated files. You are back in the processing cycle ready to execute all (sequentially coordinated or not) Main (Normal) procedures. This is the follow-up pass.
- The MXX indicates that the chained coordination miss condition is occurring at this time within the sequential file coordination cycle.

The text in the Process Cycle column indicates when sequentially coordinated and Main (Normal) procedures would be given control from VISION:Builder to perform their algorithms and/or outputs.

The MISSPASS column shows the settings in the MISSPASS flag. This value can be interrogated during the follow-up pass in the processing cycle through all Main (Normal) procedures to determine whether or not matches occurred during the sequential file coordination cycle. A value of M means at least one match occurred. A value of X means no matches occurred. A blank value indicates that the follow-up pass is not in progress. Note that this flag is blank during the chained coordination miss pass.

## Sequential File Coordination Rules

The preceding discussions and examples illustrate that the sequential file coordination cycle and the mechanisms that are available in your applications are extensive and flexible. Once you have applied your data against the various options and methods shown in the examples, you will be able to determine which techniques best fit your application requirements. By preparing some simple applications that produce reports of the information as shown in the examples, you will see precisely what effect each of the options will have on your processing flow.

As a summary, here are a number of rules that must be considered when using sequential file coordination.

- Each file must be in sequence by its entire match key.
- When there are a different number of key fields between the matching file and the coordinating files, the file with the fewest number of keys determines the number of key fields used when comparing and aligning the files. However, sequence checking on each file is still performed using all the key fields defined for the file.
- When matching to the master file, the record key value in the new master work area is used.
- If any sequentially coordinated file has a specification in the Crd entry, all sequentially coordinated files must have a specification in the Crd entry.
- Nested coordination can only be used when all files match to the master file (the Crd entry is blank). All record processing is not available with nested coordination.
- With chained coordination, only one file can match to the next file above it in the chain except at the lowest level in the chain. At the lowest level in the chain, several files can match to the next highest file; these lowest level files are processed in parallel.
- The Main (Normal) procedures are given an additional frame of reference when sequential coordination is used in an application. If a field from a sequentially coordinated file or the ECORD flag is referenced in a Main (Normal) procedure or request, it is classified as a sequentially coordinated procedure or request. This means that VISION:Builder will give those procedures or requests control for execution during the sequential file coordination cycle. Otherwise, they only get control for execution during the normal processing cycle.

Finally, remember that when the master file and all the sequentially coordinated files are at end of file, the data in their respective work areas is not available for processing. This is very important when your application contains End of File procedures. If you reference any master file or sequentially coordinated file fields in the end of file procedures, you receive invalid results and missing output. This includes any subroutine procedures or requests that might be called. During the End of File procedures, the EOF flag settings for the master file and all sequentially coordinated files are E; the ECORD flag settings are X for all the sequentially coordinated files; and the MISSPASS flag is set to a value of blank.

### Sequential File Coordination and Transaction Processing Cycle❹

In an application run involving sequential file coordination and the transaction processing cycle, the initial alignment of sequentially coordinated files with a new master record work area is performed immediately following the transaction update of the new master and prior to the first execution of After Update (type M or 3) procedures for a new master key value. Match only sequentially coordinated files are advanced to the first matching record or to the first record with a higher

match key than the new master. All records for sequentially coordinated files are compared to the new master work area to determine whether the sequentially coordinated file matches are lower than or higher than the new master. All bytes of the ECORD flag for the sequentially coordinated files are set appropriately. This in no way affects the Main (Normal) procedure and request processing of sequentially coordinated files.

If a master file record is deleted during the transaction processing cycle of the current master record, sequentially coordinated file records that match the master prior to the first execution of After Update (type M or 3) procedures no longer match during the normal processing and sequential coordination cycles.

## User Coordination

User coordination is a mechanism within VISION:Builder that allows you to control when a coordinated file is processed and how it is aligned. The timing of the read operations, the type of read operation, and the method used for alignment are completely controlled by you within the application procedures.

User-coordinated files can be any format and structure, using any access method that can be defined to and supported by VISION:Builder. This includes hierarchically structured records from IMS databases or mapped from relational tables (for example, DB2). You can also input files that contain VISION:Builder array records as coordinated files using the array definition.

Remember that any database management system or access method type that is not directly supported by VISION:Builder can be integrated using GDBI or User I/O.

User-coordinated files can be processed serially, directly, or both, depending on the commands and options specified in your application. User-coordinated files are only processed when specifically designated by commands and options within procedures; you determine when coordination processing occurs. Remember that direct-reads can only be performed when the user coordinated file's access method allows direct or random access by a key value.

Here are some common uses for user coordination:

- To read an application parameter file and store the information in temporary or working storage fields for use in subsequent procedures. This is usually done as the first task in the application. This can be accomplished using direct or serial reads.
- To access an indexed file by a secondary, alternate, or generic key and to obtain all the records from the keyed group. For example, all records in an IMS database with the same key value when the secondary index points to multiple records with the same key value. This would be accomplished using a combination of direct and serial reads.

It is important to remember that when your application requires that you completely control the timing, the reading, and the aligning of files, the user coordination method provides you with all the mechanisms you will need.

### Serial Read

Serial reads of user-coordinated files are accomplished using either the DO FORALL command or the FIND function in an ASL procedure. The RD operator is used in the fixed format syntax request coding.

- The DO FORALL command performs serial reads of records from user-coordinated files by specifying the root segment name in the command. A record is read from the user coordinated file during each iteration of the DO block. The DO block completes when the end of the user coordinated file is encountered unless an exit from the DO block is performed before end of file.
- The FIND function performs serial reads of records from user-coordinated files when *only* the root segment is specified in the function. A record is read from the user coordinated file each time the FIND function is executed in a procedure. The FIND is a conditional function that is used as a logical expression within commands.
- The RD operator in fixed format syntax request coding performs serial reads of records from user-coordinated files by coding the file qualifier (1-9) in the operation. A record is read from the user coordinated file each time the RD operator is executed in a request.

When reading user-coordinated files serially, no sequence checking is performed; records are read in physical order.

The RSTATUS flag is set after each read to indicate the result of the read. The EOF flag is also set after the read.

Refer to the *VISION:Builder ASL Reference Guide* or the *VISION:Builder Reference Summary* for the details on the commands, functions, operators, and the settings of the flags.

### Direct-Read

Direct-reads of user-coordinated files are accomplished using the FIND function in an ASL procedure and providing the key value of the record. The RE and RG operators are used in the fixed format syntax request coding. When specifying the key value for direct-reads, you can use the entire key value or a partial key value. A partial key is referred to as a generic key. Generic keys are fields defined in the file definition that start at the first position of the actual record key and have a length less than the actual record key. Direct-reads can be specified to perform a key equal read or a key greater than or equal read.

- The FIND function performs direct-reads of records from user-coordinated files when the root segment is specified and a key value is provided in the function. The key value of the record you want to read is specified in the WHERE clause of the FIND function. The key value that is specified can be the entire key value or a generic key value. Additionally, you specify how to align a record with the key value, either find a record with an equal key or find a record with a greater than or equal key. A record is read from the user

coordinated file each time the FIND function is executed in a procedure. The FIND function is a conditional function that is used as a logical expression within commands.

- The RE and RG operators in fixed format syntax request coding perform direct-reads of records from user-coordinated files by coding the file qualifier (1-9) and key values in the operation. A record is read from the user coordinated file each time the RE or RG operator is executed in a request. The RE operator is used for key equal reads and the RG operator is used for key greater than or equal reads.
- When performing direct-reads using a generic key, you must also specify the generic key field on the FILE CORDn command using the USERREAD AND GENERIC keywords for the user coordinated file. When using fixed format syntax coding, use the Additional Files (RF) statement to specify your Alternate Key field.

Direct-reads can only be performed when the user coordinated file's access method allows random or direct-access by a key value.

The RSTATUS flag is set after each read to indicate the result of the read. The EOF flag is also set after the read.

Refer to the *VISION:Builder ASL Reference Guide* or the *VISION:Builder Reference Summary* for the details on the commands, functions, operators, and the settings of the flags.

You can also use direct and serial reads in combination. First, use a direct-read operation to get positioned on a record in your file based on the key value provided for the direct-read. Next, use a serial read operation to read all the records that follow in the file that still satisfy the previous direct-read key criteria. Repeatedly execute the serial read operation until you reached the end of the group of records that meet the previous direct-read key criteria. The RSTATUS flag setting is interrogated to determine if the keys are still equal or you have reached the end of the group.

## Examples of User-Coordinated Files

The following application example illustrates how user coordination is specified, describes the logic and commands to process user-coordinated files, and shows the results based on specific input data. The application is only intended as an example of the various user coordination processing mechanisms and, therefore, contains only the minimum amount of detail for illustration purposes. The application is shown and described using the ASL (Advanced Syntax Language) coding. The complete application is shown, including the fixed format syntax coding specifications, in Appendix D, *Complete Sample Applications*.

In the example, the input master file contains only one customer record. There are four user coordinated input files (M4CORD1,2,3,4), each containing the same set of records. The user-coordinated files are all indexed and can be accessed serially

and directly. For purposes of the example, only the key values of the records are important. Figure 6-26 shows the files and the key values of the records contained in the files.

```
----- FILE KEYS VALUES -------
 ------------------------------
    MASTER1     USER CORD FILES
     FILE       NUMBER 1,2,3,4
 ------------------------------
   CUSTOMER       CUSTOMER AND
    NUMBER        ORDER NUMBER
 ------------------------------
    00002          0000110001
                   0000110004
                   0000110005
                   0000220003
                   0000220005
                   0000330001
                   0001001001
                   0001001004
                   0002002003
```

Figure 6-26 Key Values of the Records in the Master File and the user-coordinated files

Look at the file definitions that you are using in the application example. Figure 6-27 shows the definitions that are used for the master file in the example. The master file is called MASTER1.

```
                                    DETAILED GLOSSARY BY LOCATION FOR                                  GLOSSARY PAGE      1
                                        FILE DEFINITION - MASTER1
DATE CATALOGED - 11/03/01 13:20:19                                                DATE LAST UPDATED - 11/03/01 13:20:19
CATALOGED BY -                                                                    LAST UPDATED BY -
EXPIRATION DATE -                                                                 DATE LAST USED -
FILE IDENTIFICATION =                                                             RECORD FORMAT = FIXED UNBLOCKED
NUMBER OF SEGMENTS IN FILE = 1                                                    RECORD SIZE = 50
NUMBER OF FIELDS IN FILE = 2                                                      BLOCK SIZE = 50

                                     ******************************
                                     *    SEGMENT   1, LEVEL 1     *
                                     ******************************
SEGMENT OCCURS N TIMES = 1                                        KEY FIELD 1 = CUSTNUM    TYPE = C   LENGTH = 5
SEGMENT SIZE = 50
NUMBER OF FIELDS IN SEGMENT = 2
-------------------------------------------------------------------------------------------------------------------------------
FIELD   FIELD   FIELD    FIELD   FIELD    DEC    CNT FIELD  EDIT CODES     EDIT    OUTPUT   LINE  DATA BASE NAME/
NAME    TYPE   LOCATION  LENGTH  RNDING  PLACES  FOR SGMT   ( ) ( ) ( )   LENGTH   WIDTH     NO   *** COLUMN HEADING ***
-------------------------------------------------------------------------------------------------------------------------------
CUSTNUM   C        1       5                                                 5       8
                                                                                          1    ***    CUSTOMER    ***
                                                                                          2    ***     NUMBER     ***
DATA      C        6      45                                                45      45
                                                                                               ***                ***
                                                                                          2    ***   MSTR DATA    ***
```

Figure 6-27 File Definition Showing the File Structure and Format of the Customer File

The MASTER1 file definition shows that there is only one segment, the root segment, in the file. The root segment is named CUSTSEG. There are two fields defined: CUSTNUM is the key field and DATA defines the remainder of the segment.

Figure 6-28 shows the definitions that are used for all the user coordinated input files in the example. The file is called CORDFILE. Because all the user-coordinated files are identical in structure and format, one definition is all that is needed.

```
                                    DETAILED GLOSSARY BY LOCATION FOR                                  GLOSSARY PAGE      1
                                        FILE DEFINITION - CORDFILE
DATE CATALOGED - 11/03/01 13:20:19                                          DATE LAST UPDATED - 11/03/01 13:20:19
CATALOGED BY -                                                              LAST UPDATED BY -
EXPIRATION DATE -                                                           DATE LAST USED -
FILE IDENTIFICATION =                                                       RECORD FORMAT = KEY SEQUENCED VSAM
NUMBER OF SEGMENTS IN FILE = 1                                              RECORD SIZE = 50
NUMBER OF FIELDS IN FILE = 5                                                BUFFER SIZE = 50

                                    ********************************
                                    *    SEGMENT   1, LEVEL 1     *
                                    *   SEGMENT NAME = ORDRSEG    *
                                    ********************************
SEGMENT OCCURS N TIMES = 1                                  KEY FIELD 1 = CUSTORDR   TYPE = C   LENGTH = 10
SEGMENT SIZE = 50
NUMBER OF FIELDS IN SEGMENT = 5
SEGMENT KEY CHARACTERISTICS = U
-------------------------------------------------------------------------------------------------------------------------------
FIELD   FIELD   FIELD    FIELD   FIELD    DEC    CNT FIELD  EDIT CODES     EDIT    OUTPUT   LINE  DATA BASE NAME/
NAME    TYPE   LOCATION  LENGTH  RNDING  PLACES  FOR SGMT   ( ) ( ) ( )   LENGTH   WIDTH     NO   *** COLUMN HEADING ***
-------------------------------------------------------------------------------------------------------------------------------
CUSTORDR  C        1       10                                              10       12
                                                                                          1    ***   CUSTOMER &   ***
                                                                                          2    ***  ORDER NUMBER  ***
CUSTNUM   C        1        5                                               5        6
                                                                                          1    ***     CUSTNO     ***
                                                                                          2    ***      ONLY      ***
CUST04    C        1        4                                               4        7
                                                                                          1    ***    CUSTNO      ***
                                                                                          2    ***    POS 1-4     ***
ORDERNUM  C        6        5                                               5        7
                                                                                          1    ***    ORDERNO     ***
                                                                                          2    ***     ONLY       ***
DATA      C       11       40                                              40       40
                                                                                               ***                ***
                                                                                          2    ***   CORD DATA    ***
```

Figure 6-28 File Definition Showing the File Structure and Format of CORDFILE

The CORDFILE file definition shows that there is only one segment, the root segment, in the file. The root segment is named ORDRSEG. There are five fields defined. CUSTORDR is the entire 10-byte key field for the root segment and, therefore, the record. This entire key contains a CUSTNUM in the first 5 bytes. Also, the first 4 bytes are defined as CUST04. The CUSTNUM and CUST04 data fields are used as generic keys in the application example. ORDERNUM is a data field that is also the last 5 bytes of the entire key field. Finally, DATA defines the remainder of the segment

Now that the files are defined, you can take a look at the application.

The application contains one procedure that produces five reports that use the six input files. The application is structured so that a single MAIN procedure initiates the reads of the user-coordinated files using DO FORALL and IF FIND commands. REPORT commands are coded to produce the various reports.

The six files in the application are the master and five user coordinated input files. Figure 6-29 shows the Run Control commands that detail the files in the application.

```
CONTROL NAME F160 TERM LISTGEN
FILE MASTER INPUT MASTER1
FILE CORD1 CORDFILE USERREAD
FILE CORD2 CORDFILE USERREAD
FILE CORD3 CORDFILE GENERIC CUSTNUM
FILE CORD4 CORDFILE GENERIC CUST04
FILE CORD5 CORDFILE USERREAD
FILE REPORT
```

Figure 6-29 User Coordination Examples Application: Run Control

**Note:** For a complete version of this example, see User Coordination Examples Application: Run Control – Part 1 of 7 on page D-53.

The application has a master file (M4OLD) that is defined by the definition called Master1. There are five coordinated input files (M4CORD1, M4CORD2, M4CORD3, M4CORD4, and M4CORD5) that are all defined by the definition called CORDFILE. Because each coordinated file is identical in format and structure, the same definition is used for all the files.

The FILE CORDn commands specify to VISION:Builder that these coordinated files are to be read and aligned as user coordinated input files. The USERREAD and GENERIC keywords indicate user coordinated processing.

- The USERREAD keyword specifies that the file is read serially in physical order or it is read directly using the *entire key* for aligning to a record in the file.
- The GENERIC keyword specifies that the file is read directly using a *generic key* specified for the file. Generic keys are fields defined in the file definition as data fields that start at the first position of the actual record key and have a length less than the actual record key.

In this application, you are reading the user-coordinated files as follows:

| | |
|---|---|
| M4CORD1 | Serially read all records in the file. |
| M4CORD2 | Directly read a record using the entire key aligning on a record with a specific key value. |
| M4CORD3 | Directly read a record using a generic key that is the CUSTNUM portion of the actual key and serially read all records that have the same generic key value (CUSTNUM) until all records in the generic group are processed. |
| M4CORD4 | Directly read a record using a generic key that is the first four positions of the actual key and serially read all records that have the same generic key value (in the first four positions) until all records in the generic group are processed. |
| M4CORD5 | Serially read all records in the file. |

The application procedure contains DO FORALL and IF FIND commands that initiate user coordinated file processing. The points within the procedure are commented to indicate which action is performed and which report is produced.

The MAIN procedure regains control after each master file record is read. Control flows to each of the DO FORALL and IF FIND commands. After all DO FORALL, IF FIND, and REPORT commands have been executed, the MAIN procedure is done and another master file record is read. This continues until all master records are processed. In the example, the master file contains only one record and, therefore, you will only execute the MAIN procedure one time.

For this example, each DO FORALL or IF FIND command handles the reading, aligning, processing, and reporting of only one of the user-coordinated files.

Figure 6-30 shows the commands for the processing of the CORD FILE 1.

```
;
; Read CORD FILE 1 "SERIALLY" and Report ALL Records
;
DO FORALL 1.ORDRSEG
   REPORT O.CUSTNUM 1.CUSTORDR F.RSTATUS
    TITLE 'SERIALLY READ CORD FILE'
    TITLE 'LIST FILE KEYS AND READ STATUS FLAG'
   END REPORT
END DO
```

Figure 6-30 Procedure Coding for Serial Read of CORD FILE 1

**Note:** For a complete version of this example, see User Coordination Examples Application: Coding for Serial Read of CORD FILE 1 – Part 3 of 7 on page D-54.

This portion of the procedure reads the M4CORD1 user coordinated file serially in physical order and outputs a report of information from the records read.

The DO FORALL 1.ORDRSEG command initiates the serial read and establishes a repetitive DO block. The DO block contains a set of statements that are repetitively executed. The END command designates the end of statements in the DO block. For each iteration of the DO block, a record (the root segment ORDRSEG) from the M4CORD1 (1. qualifier) file is read serially.

For each successful read, all the statements in the DO block are executed. In this case, the REPORT statement is used to produce an output report. The DO block repeats this for all records (all occurrences of the root segment) in the M4CORD1 user coordinated file (that is, until end of file). When the DO block completes, the statements following the END command are executed.

**Note:** You could have specified DO WHILE FIND(1.ORDRSEG) instead of the DO FORALL 1.ORDRSEG and accomplished the same thing. Because the FIND function is conditional, the DO block would repeat as long as each FIND function was successful. When end of file is encountered, the FIND function would be unsuccessful and the DO block complete.

The REPORT command produces a report containing three fields:

O.CUSTNUM Master file record key field from the old master work area.

| | |
|---|---|
| 1.CUSTORDR | M4CORD1 record key field from the M4CORD1 work area. |
| F.RSTATUS | System flag RSTATUS that shows the setting as a result of the user coordinated file read. |

Using the data shown in Figure 6-26 on page 6-49 and the commands shown in Figure 6-30, the report will output the information shown in Figure 6-31.

```
                                    SERIALLY READ CORD FILE                                    PAGE 1
                                LIST FILE KEYS AND READ STATUS FLAG
-------------------------------------------------------------------------------------------------------------------------
 CUSTOMER   CUSTOMER &   RSTATUS
  NUMBER   ORDER NUMBER
-------------------------------------------------------------------------------------------------------------------------
  00002     0000110001    KEQL
  00002     0000110004    KEQL
  00002     0000110005    KEQL
  00002     0000220003    KEQL
  00002     0000220005    KEQL
  00002     0000330001    KEQL
  00002     0001001001    KEQL
  00002     0001001004    KEQL
  00002     0002002003    KEQL
```

Figure 6-31 Output Information Produced by the Serial Read of CORD FILE 1

All records are reported from M4CORD1 (CUSTORDR key field) while the one master record (CUSTNUM key field) is available in the old master work area. The RSTATUS flag value is KEQL after each successful read. When end of file is encountered, the EOF flag is set and the RSTATUS is set to EGRP.

Figure 6-32 shows the procedure coding for DIRECT processing of CORD FILE 2.

```
;
; Read CORD FILE 2 "DIRECTLY" and Report "EQUAL" Record
;
IF FIND(2.ORDRSEG WHERE 2.CUSTORDR = '0000220005')
   REPORT O.CUSTNUM 2.CUSTORDR F.RSTATUS
    TITLE 'DIRECTLY READ CORD FILE'
    TITLE 'LIST FILE KEYS AND READ STATUS FLAG'
   END REPORT
END IF
```

Figure 6-32 Procedure Coding for DIRECT Processing of CORD FILE 2

**Note:** For a complete version of this example, see User Coordination Examples Application: Coding for DIRECT Processing of CORD FILE 2 – Part 4 of 7 on page D-54.

The procedure reads the M4CORD2 user coordinated file directly using the entire key (CUSTORDR), aligning on a record that contains a specific key value (0000220005). A report is output containing the information from the record read.

CORD FILE 2 is processed using an IF command that establishes an IF block. The FIND(2.ORDRSEG WHERE 2.CUSTORDR = 0000220005) function is the logical expression for the IF block and it initiates a direct-read. The IF block contains a set of statements that are executed when the direct-read finds a record in the M4CORD2 user coordinated file (that is, the logical expression is true). The END command designates the end of the statements in the IF block.

For each execution of the IF command, the FIND function attempts to directly read a record (the root segment ORDRSEG) from the M4CORD2 (2.qualifier) file that has a key value equal to 000022005. When the find is successful, the statements in the IF block are executed. In this case, a REPORT command is used to produce an output report.

Because the IF block is not within a repeating DO block, it attempts to find the record only once. If the record is found, the report output is prepared. If there is no record with the provided key value on the user coordinated file, the procedure goes to the END of the IF block. In either case, the statements following the END command are executed.

**Note:** The key value provided in the FIND function WHERE clause does not have to be a constant, it can be a value in a field.

The REPORT command produces a report containing three fields:

| | |
|---|---|
| O.CUSTNUM | Master file record key field from the old master work area. |
| 2.CUSTORDR | M4CORD2 record key field from the M4CORD2 work area. |
| F.RSTATUS | System flag RSTATUS that shows the setting as a result of the user coordinated file read. |

Using the data shown in Figure 6-26 on page 6-49 and the commands shown in Figure 6-32, the report outputs the information shown in Figure 6-33.

```
                                    DIRECTLY READ CORD FILE                                    PAGE 1
                                LIST FILE KEYS AND READ STATUS FLAG
-------------------------------------------------------------------------------------------------------------------------
 CUSTOMER   CUSTOMER &   RSTATUS
  NUMBER   ORDER NUMBER
-------------------------------------------------------------------------------------------------------------------------
  00002     0000220005    KEQL
```

Figure 6-33 Output Information Produced by the Direct Read of CORD FILE 2

What you get is the record reported from M4CORD2 (CUSTORDR key field) that has the key value equal to the key value provided in the FIND function WHERE clause. The key of the one master record (CUSTNUM key field) is also reported from the old master work area. The RSTATUS flag value is KEQL after a successful direct-read for an equal key value. If there is no record in the M4CORD2 file with a record key equal to the provided value, report output is bypassed (that is, no data selected).

If you need to perform tasks in your application when no record is found, you have several options. For example, using the ELSE command with the IF command allows you to execute a group of statements when no record is found, meaning the FIND function was unsuccessful and the logical expression was false. You could also interrogate the RSTATUS flag that gets set to NREC when no record is found on a direct-read attempt. Based on the check of the RSTATUS flag, you could go to statements in your procedure to perform whatever tasks are needed.

The next portion of the application procedure illustrates how to perform generic key processing. Figure 6-34 shows the procedure coding.

```
;
; Read CORD FILE 3 "GENERICALLY' and Report Records within the
; CUSTNUM Group; having the same customer number
;
IF FIND(3.ORDRSEG WHERE 3.CUSTORDR >= O.CUSTNUM)
  DO WHILE F.RSTATUS = 'KEQL' OR F.RSTATUS = 'KGTR'
    REPORT O.CUSTNUM 3.CUSTORDR F.RSTATUS
     TITLE 'GENERIC READ CORD FILE BY CUSTOMER NUMBER'
     TITLE 'LIST FILE KEYS AND READ STATUS FLAG'
    END REPORT
    IF FIND(3.ORDRSEG)
    END IF
  END DO
END IF
```

Figure 6-34 Procedure Coding for Generic Key Processing

**Note:** For a complete version of this example, see User Coordination Examples Application: Coding for Generic Key Processing of CORD FILE 3 – Part 5 of 7 on page D-54.

The procedure reads the M4CORD3 user coordinated file directly using a generic key and serially reads all records that have the same generic key. The generic key is specified on the FILE CORD3 command (see Figure 6-29 on page 6-51). In this case, the generic key is the CUSTNUM portion of the entire key CUSTORDR. This procedure directly aligns on a record that contains a value in the first 5 bytes of the key field that is equal to the CUSTNUM from the master file. Because you want to report all records from the M4CORD3 file that have a value in the key field that is the same CUSTNUM as the master file record, the procedure serially reads all records with the same generic key and outputs the information from the records read.

CORD FILE 3 is processed using an IF command that establishes an *outer* IF block. The FIND(3.ORDRSEG WHERE 3.CUSTORDR = O.CUSTNUM) function is the logical expression for the *outer* IF block and it initiates a direct-read. The *outer* IF block contains a set of statements that are executed when the direct-read finds a record in the M4CORD3 user coordinated file (that is, the logical expression is true). The *outer* IF command with the FIND function attempts to directly read a record (the root segment ORDRSEG) from the M4CORD3 (3. qualifier) file that has a generic key value (the first 5 bytes) equal to the CUSTNUM from the old master file (O. qualifier) work area. When the FIND function is successful, all the statements in the *outer* IF block are executed.

After successfully aligning to a record with a generic key equal to the master CUSTNUM, the procedure executes the statement with a DO command. The DO WHILE F.RSTATUS = 'KEQL' establishes a DO block within the *outer* IF block.

The DO block contains statements that are repetitively executed while the RSTATUS flag is equal to KEQL (that is, as long as you find and continue to find M4CORD3 records that have a generic key equal to the master CUSTNUM). The REPORT command is used to produce an output report of the M4CORD3 information read.

Next, an *inner* IF FIND(3.ORDRSEG) initiates the serial reads on the M4CORD3 (3.ORDRSEG) file. Because you have just executed a successful direct-read using a generic key against the M4CORD3 file, the serial read starts by reading the next physical record following the direct-read alignment. The next statement is the END command for the *inner* IF block.

Because the REPORT command and the *inner* IF block are both within the DO block, they are repeatedly executed while the serial reads continue to find M4CORD3 records with an equal generic key value (RSTATUS = 'KEQL').

When you read a record that does not equal the generic key value, you are at the end of the generic group and the RSTATUS is set to EGRP. This means that the DO block is complete. The END command is executed for the DO block, followed by the END command for the *outer* IF block.

The REPORT command produces a report containing three fields:

| | |
|---|---|
| O.CUSTNUM | Master file record key field from the old master work area. |
| 3.CUSTORDR | M4CORD3 record key field from the M4CORD3 work area. |
| F.RSTATUS | System flag RSTATUS that shows the setting as a result of the user coordinated file read. |

Using the data shown in Figure 6-26 on page 6-49 and the commands shown in Figure 6-34, the report outputs the information shown in Figure 6-35.

```
                                    GENERIC READ CORD FILE BY CUSTOMER NUMBER                    PAGE 1
                                      LIST FILE KEYS AND READ STATUS FLAG

 CUSTOMER    CUSTOMER &   RSTATUS
  NUMBER    ORDER NUMBER
-------------------------------------------------------------------------------------------------------

   00002      0000220003    KEQL
   00002      0000220005    KEQL
```

Figure 6-35 Output Information Produced by Generic Read of CORD FILE 3

The report consists of two records from M4CORD3 (CUSTORDR key field) that have a generic key value (00002 in the first 5 bytes) equal to the CUSTNUM key value (00002) from the one master file record in the old master work area. The RSTATUS flag value is KEQL after a successful direct-read for an equal generic key value, as well as all subsequent successful serial reads for the same generic key value.

If there is no record in the M4CORD3 file with a generic key equal to the master file key, the *outer* IF contains a false condition from the FIND function and the DO block, the *inner* IF block is not executed, and report output is bypassed (that is, no data selected).

If you need to perform tasks in your application when no record is found, you have several options. For example, using the ELSE command with the *outer* IF command allows you to execute a group of statements when no record is found, meaning the FIND function is unsuccessful and the logical expression is false. You could also

interrogate the RSTATUS flag that gets set to NREC when no record is found on a direct-read attempt. Based on the check of the RSTATUS flag, you could go to statements in your procedure to perform whatever tasks are needed.

The last portion of the procedure is also an example of how to perform generic key processing. Figure 6-36 shows the procedure coding for processing CORD FILE 4.

```
;
; Read CORD FILE 4 "GENERICALLY' and Report Records where the first
; four positions in the CUSTNUM match the same in the CORD FILE
;
IF FIND(4.ORDRSEG WHERE 4.CUSTORDR >= PF(N.CUSTNUM,1,4))
  DO WHILE F.RSTATUS = 'KEQL' OR F.RSTATUS = 'KGTR'
    REPORT O.CUSTNUM 4.CUSTORDR F.RSTATUS
     TITLE 'GENERIC READ CORD FILE BY CUSTOMER NUMBER'
     TITLE 'LIST FILE KEYS AND READ STATUS FLAG'
    END REPORT
    IF FIND(4.ORDRSEG)
    END IF
  END DO
END IF
```

Figure 6-36 Procedure Coding for Processing CORD FILE 4

**Note:** For a complete version of this example, see User Coordination Examples Application: Coding for Processing of CORD FILE 4 – Part 6 of 7 on page D-55.

The procedure works just like the previous portion of the procedure. The only difference is that you are using only the first 4 bytes of the user coordinated file as a generic key.

The procedure reads the M4CORD4 user coordinated file directly using a generic key and serially reads all records that have the same generic key. The generic key is specified on the FILE CORD4 command (see Figure 6-29). In this case, the generic key is the CUST04 portion of the entire key CUSTORDR — only the first 4 bytes. The procedure directly aligns on a record that contains a value in the first 4 bytes of the key field that is equal to the first 4 bytes of the CUSTNUM field from the master file. Because you want to report all records from the M4CORD4 file that have a key with the same first 4 bytes as the master file record key, the procedure serially reads all records with the same generic key and calls a report request to output the information from the records read.

CORD FILE 4 is processed using an IF command that establishes an *outer* IF block. The FIND(4.ORDRSEG WHERE 3.CUSTORDR = PF(O.CUSTNUM,1,4) function is the logical expression for the *outer* IF block and it initiates a direct-read. The *outer* IF block contains a set of statements that are executed when the direct-read finds a record in the M4CORD4 user coordinated file (that is, the logical expression is true). The *outer* IF command with the FIND function attempts to directly read a record (the root segment ORDRSEG) from the M4CORD4 (4. qualifier) file that has a generic key value (the first 4 bytes) equal to the first 4 bytes of CUSTNUM from the old master file (O. qualifier) work area. When the FIND function is successful, all the statements in the *outer* IF block are executed.

After you have successfully aligned to a record with a generic key equal to the first 4 bytes to the master CUSTNUM, the DO WHILE F.RSTATUS = 'KEQL' establishes a DO block within the *outer* IF block.

The DO block contains statements that are repetitively executed while the RSTATUS flag is equal to KEQL (that is, as long as you find and continue to find M4CORD4 records that have a generic key equal to the first 4 bytes of the old master CUSTNUM field). The first statement in the DO block is a REPORT command that produceS an output report of the M4CORD4 information read.

Next, an *inner* IF FIND(4.ORDRSEG) initiates the serial reads on the M4CORD4 (4.ORDRSEG) file. Because you have just executed a successful direct-read using a generic key against the M4CORD4 file, the serial read starts by reading the next physical record following the direct-read alignment. The next statement is the END command for the *inner* IF block.

Because the REPORT command and the *inner* IF block are both within the DO block, they are repeatedly executed while the serial reads continue to find M4CORD4 records with an equal generic key value (RSTATUS = 'KEQL').

When you read a record that does not equal the generic key value, you are at the end of the generic group and the RSTATUS is set to EGRP. This means that the DO block is complete. The END command is executed for the DO block, followed by the END command for the *outer* IF block.

The REPORT command produces a report containing three fields:

| | |
|---|---|
| O.CUSTNUM | Master file record key field from the old master work area. |
| 4.CUSTORDR | M4CORD4 record key field from the M4CORD4 work area. |
| F.RSTATUS | System flag RSTATUS that shows the setting as a result of the user coordinated file read. |

Using the data shown in Figure 6-26 on page 6-49 and the commands shown in Figure 6-36, the report outputs the information shown in Figure 6-37.

```
                                    GENERIC READ CORD FILE BY CUSTOMER NUMBER                    PAGE 1
                                      LIST FILE KEYS AND READ STATUS FLAG

 CUSTOMER   CUSTOMER &   RSTATUS
  NUMBER   ORDER NUMBER

   00002     0000110001    KEQL
   00002     0000110004    KEQL
   00002     0000110005    KEQL
   00002     0000220003    KEQL
   00002     0000220005    KEQL
   00002     0000330001    KEQL
```

Figure 6-37 Output Information Produced by Generic Read of CORD FILE 4

**Note:** In the example, the key alignment is based on an equal key. The operator in the WHERE clause of the FIND function is always “=” for equal. To align on greater than or equal keys, specify the “>=” or “GE” operator in the WHERE clause of the FIND function. The points within your application where the reading and aligning occurs depends on what options you specify and when you execute the appropriate commands. VISION:Builder does the detail reading and, if specified, the detail aligning by key, but only under your direction.

Six records are reported from M4CORD4 (CUSTORDR key field) that have a generic key value (0000 in the first 4 bytes) equal to the first 4 bytes of the CUSTNUM key value (00002) from the one master file record in the old master work area. The RSTATUS flag value is KEQL after a successful direct-read for an equal generic key value, as well as all subsequent successful serial reads for the same generic key value.

If there is no record in the M4CORD4 file with a generic key equal to the master file key, the *outer* IF contains a false condition from the FIND function and the DO block, the *inner* IF block is not executed, and the report output is bypassed (that is, no data selected).

If you need to perform tasks in your application when no record is found, you have several options. For example, using the ELSE command with the *outer* IF command allows you to execute a group of statements when no record is found, meaning the FIND function is unsuccessful and the logical expression is false. You could also interrogate the RSTATUS flag that gets set to NREC when no record is found on a direct-read attempt. Based on the check of the RSTATUS flag, you could go to statements in your procedure to perform whatever tasks are needed.

### Fixed Format Syntax Listing of User-Coordinated Files Example

Appendix D, *Complete Sample Applications* shows the complete VISION:Builder source listing and reports produced for the example. The fixed format syntax coding is shown on the source listing. Notice the RD and RE operators that activate the reading and aligning mechanisms of user coordination processing; there are interrogations of the EOF and RSTATUS flags to determine the status of the operations. The serial reads (RD) operations are repeated using request back branching. Refer to the *VISION:Builder Reference Summary* for more details on the statements and entries in the example. Also, refer to Chapter 11, *Requests* in this manual for a discussion on requests.

## Source Statements

Every VISION:Builder engine run requires input specifications to initiate actions by the engine. These input source statement specifications are known as the M4INPUT. The input file or data set characteristics are sequential fixed length records that are 80 bytes long.

In application processing runs, the M4INPUT contains the source statements that specify your application processing requirements. In definition/maintenance runs, the M4INPUT contains the source statements that define or change your file, transaction,❹ table, or array definitions that are cataloged and stored on the common library. In common library utility runs, the M4INPUT contains the source statements that specify the utility operations to be performed.

Chapter

# 7 Output File Processing

VISION:Builder automatically handles all input, output, and interfile matching of your application's data files for you. In order to do this, VISION:Builder has a number of files – input and output – from which you can choose, according to the requirements of your application.

This chapter discusses the output files available from VISION:Builder applications and describes the various types of output files, their normal usage, and how they are processed. As you design your application, you have several options for producing and controlling output information from your application. This chapter discusses the files that can contain your output information. The description of output reports is described in Chapter 9, *Reporting*.

The following table shows the types of output files and their usual usage in VISION:Builder applications

| VISION:Builder Name | DTF/DD Name | Usage in a VISION:Builder Application Program | Comments |
|---|---|---|---|
| New Master File | M4NEW❹ | The updated version of the master file. | M4NEW is the output master file. |
| Report Files | M4REPO<br>M4REPn<br>M4REPI | Temporarily holds the individual (detail level) report command records and the report data records for the various reports generated by the application program. | See the *VISION:Builder for OS/390 Environment Guide* (OS/390 users) or the *VISION:Builder for VSE Environment Guide* (VSE users).<br><br>n = 2 through 9 |
| Report Summary Files | User Specified | Output files that contain the summary information from selected reports. This would be used as input in subsequent applications. | Use is optional. |
| Subfiles | M4SUBFn | Output files that can be any subset of data available to the application program, including selected entire records from any application input file. | n = 0 through 9<br><br>May contain array records. |

| VISION:Builder Name | DTF/DD Name | Usage in a VISION:Builder Application Program | Comments |
|---|---|---|---|
| Rejected Transactions File¼ | M4REJCT❹ | Transaction records that have been rejected. Output for correction and retry in a subsequent run. | Use is optional. |
| Deleted Master File Records¼ | M4AUDIT❹ | Audit trail of deleted master file records. | Use is optional. |
| Source and Report Listings File | M4LIST | Listing of decoded source statements, reports from application runs, glossaries from definition/maintenance runs, and library utility run reports. | All printed output is normally written to this DTF/DD name from all runs. |
| Application Reports (Alternate List File for Report Listings) | M4LIST1 (User Specified) | Application reports that are directed to an alternate list file (DTF/DD). The DTF/DD name is M4LIST1 or a unique name supplied in the application. | Reports can be directed to user specified DTF/DD destinations. |
| Alternate Report Output Methods | User Specified | Use to prepare data for input to other tools, processes, and platforms. | Most of the alternate output formats are industry-standard, open formats. |

In some cases, VISION:Builder understands the format and attributes of the output file because it is based entirely on an input file that is associated with a file definition. In other cases, the format and attributes are determined by direct specifications in the application or by using the standard system characteristics or defaults for the designated output file.

VISION:Builder performs all the file handling chores for you automatically for all the access methods and database managers that are directly supported. In these situations, VISION:Builder performs:

- All opens and closes.
- All writes.
- All error trapping, recovery (if possible), or graceful termination.

For access methods and database management systems that are not directly supported by VISION:Builder, you can use either GDBI (General Database Interface) or User I/O to invoke your accessing module. User I/O is documented in the *VISION:Builder for OS/390 Environment Guide* (OS/390 users) or the *VISION:Builder for VSE Environment Guide* (VSE users). Briefly, VISION:Builder invokes your routine each time it needs to perform an output function for the designated files. Your routine performs the requested function and passes data from VISION:Builder in the format described in the appropriate VISION:Builder output specifications. After the output I/O, your VISION:Builder application behaves as if VISION:Builder had performed the output functions.

GDBI is described in the GDBI Programming Manual. It interacts with VISION:Builder in the same manner as User I/O, except it functions at the segment level instead of the record level.

The access methods that VISION:Builder supports directly include:

- VSAM
- Sequential
- ISAM

The database management systems include:

- IMS
- DB2

## Master File

The master file is the primary file for all VISION:Builder applications. The term master signifies only that the file so designated is the primary file in the application run. There is no significance to the term with respect to data or information attributes of the file. There are two work areas, old and new, that are set up to accommodate the data in the master file. The old master work area contains the data that is input to the application from the old master file and is only available when an old master file is present in the run. The new master work area initially contains a copy of the old master data and is available for updating by the transaction input data❹ and your application procedures and requests. The new master work area is always available to a VISION:Builder application.

The new master work area can be output.❹ When the new master work area is output, there are two choices as to where the data is written. The new master data can be output to an entirely new file, data set, or database. Alternatively, the new master record data can be output to the same physical location that contained the original old master input record. This is known as update-in-place.❹ See Chapter 6, *Input File Processing* for additional information on update-in-place.

The format and structure of the new master output file or database is always the same as the old master input file or database.

When the M4NEW file❹ is specified by the ASL FILE MASTER INPUT statement or the Run Control (RC) fixed format statement for your application, the new master record is output to a new file or database. The processing is in output only mode. The data set or database and the access method should be set up to handle the output only processing mode.

When update-in-place❹ is specified, it is specified by the ASL FILE MASTER UPDATE statement (or the RC statement) and no M4NEW is specified. The output of the new master data is processed in update mode and directed to the same physical location as the M4OLD input. The M4OLD file or database is processed in read/write (update) mode. The data set or database and the access method should be set up to handle this processing mode.

Finally, if the input old master file is being processed (read) directly,❹ the output new master will also be processed (written) directly.

Note that the processing options used for the old master (M4OLD) have an effect on how the new master (M4NEW) is processed when you specify a new master output file.❹ Chapter 6, *Input File Processing* has more details on the processing of the old master file.

Figure 7-1 shows an example of a Run Control command group with an old and new master file specified.

```
CONTROL
FILE MASTER INPUT, NAME CUSTMAST
FILE MASTER OUTPUT
```

Figure 7-1 Run Control Command Group with Old Master (M4OLD) and New Master (M4NEW) Entries

**Note:** For the expanded source listing of this example, see Run Control Command Group with Old Master (M4OLD) and New Master (M4NEW) Entries on page D-60.

# Report Files

Report files contain all the information and data needed to produce your application's output reports. These files are processed sequentially and contain variable length records. The format and structure of report files are of an internal construct that has been designed for use by VISION:Builder. The record and block size values used for report files are standard system values. These standard values are contained in the M4PARAMS module. The M4REPO block size value from M4PARAMS is also displayed on the source listing banner page. To change standard system defaults, see the *VISION:Builder for OS/390 Installation Guide* (OS/390 users) or the *VISION:Builder Installation Manual for VSE* (VSE users) for details on how to update the M4PARAMS module.

A report file contains a file header record that identifies it as a VISION:Builder report file. For each report in your application, the report file contains a set of report command records along with report data records. The report command records contain all the information, titles, headings, edit patterns, and so on, needed to format your report. The report command records also contain the detailed instruction steps for producing the formatted output.

The report data records contain the detail data values for each field specified and selected for the output report. All the records in the report file contain a sort key field. This allows the file to be sorted into the correct order for reporting. The sort key contains report identifiers, record type identifiers, and the data values to be used for sequencing the detail report data.

The report file is produced in the processing phase of an application run. It is output in the order that the input files and data were processed. Once the report file is sorted into sequence by its sort key, it is processed as input to the report phase, which produces the formatted report output.

Usually, all your application report outputs use the M4REPO report file. In a single-step application run, the M4REPO report file is used to temporarily store only the report command records during the processing phase. The report file data records are passed directly to the sort. At the end of the processing phase, the report command records stored on the M4REPO report file are passed to the sort. The sort returns the sorted report file for input to the report phase, which produces the report output.

In a three-step application run, the processing step writes all the report file records to the M4REPO file. This file is input to the sort and the sort output becomes the sorted report file (M4REPI). The M4REPI is input to the application report step and the formatted report output is produced.

With VISION:Builder, you have an option that allows you to direct the report file command and data records for individual reports to alternate report files. These alternate report files are known as M4REP2 through M4REP9. You specify an alternate report file on the ASL File Report statement or an Additional Files (RF) statement. You can indicate that an individual report be directed to the alternate report file by using an ASL REPORT statement with a FILE command or an Output Specification (En) statement. More than one individual report can be directed to the same alternate report file.

Because all the information and data needed to produce a given formatted report is contained on the report file, alternate report files can be sorted and have the application report phase run performed at any time (that is, scheduled separate from the application processing run). Remember, the alternate report file contains information to produce a report, not the actual formatted report.

Figure 7-2 and Figure 7-3 show an example of the specifications needed to direct a report to an alternate report file for subsequent sorting and reporting.

```
FILE REP2  ALTREP2
```

Figure 7-2 ASL FILE Statement Specifying an Alternate Report File (M4REP2)

**Note:** For the expanded source listing of this example, see on page D-60.

Notice that the entry for the alternate report file M4REP2 does not use a qualifier. It is not referenced by qualifier; instead, you supply a file name to use as a reference. In this example, ALTRPT2 is supplied. The ddname chosen is M4REP2. The values M4REP2 through M4REP9 are valid for alternate report files.

```
REP1: REPORT COLUMNS T.TEMP1  FILE ALTREP2
END REPORT
```

Figure 7-3 Report Definition Showing Alternate Report File Specification

**Note:** For the expanded source listing of this example, see Run Control Command Group with Old Master (M4OLD) and New Master (M4NEW) Entries on page D-60.

You indicate the name of the alternate report file where you want this report's information to be output. The name ALTREP2 associates the specific alternate report file with the the one named on the FILE REPn statement in the Run Control command group.

# Subfiles

A subfile is an output data file produced by your application. The subfile can be any format and structure supported by VISION:Builder. The subfile is processed in output mode. The opens, closes, and output writes are performed automatically when subfile output is specified.

The subfile output record can be a copy of an entire record from any other input file work area (including the new master work area) that is available in your application. The subfile output can also be a newly created record that consists of selected fields from any work area available in your application.

With VISION:Builder, you can output up to 138 subfiles from a single application. There are 10 standard subfile outputs (M4SUBF0 through M4SUBF9) available using the combination of a FILE M4SUBFn and EXTRACT FILE command. There are 128 extended subfile outputs available using only an EXTRACT DDNAME, EXTRACT DBDNAME, or EXTRACT TABLE command.

In the following discussions of subfiles, the only difference between Standard subfiles and Extended subfiles is use of the FILE SUBFn command. There are three keywords on the FILE SUBFn that are unique to Standard subfiles.: PASSWORD, IOPLUGIN and DELETESEGS. See the *VISION:Builder ASL Reference Guide* for details regarding these keywords. Other than the FILE SUBFn command specifications, the handling and capabilities available for Standard and Extended subfiles are identical.

When you specify a subfile output, you can have VISION:Builder automatically generate the file definition statements that describe the format, structure, and content of the subfiles in your application. These file definition statements are output to a subfile designated in your application. The file definition statements are suitable for input to a VISION:Builder definition/maintenance run for cataloging into your common library.

Standard subfiles are specified using the FILE SUBn statements in the Run Control command group and the EXTRACT statement in the Subfile Output command group. In fixed format syntax, the Additional Files (RF) and Output Specification (En, Rn) statements are used. Figure 7-4 shows a Run Control command group containing subfile specifications.

```
FILE SUBF1 NAME CUSTSUB
FILE SUBF2 NAME ITEMSUB
FILE SUBF0 NAME DEFS4SUB, AUTODEF ; This file holds file definitions
                                  ; generated for subfiles
```

Figure 7-4 FILE Statements Showing Subfile Specification Entries

**Note:** For the expanded source listing of this example, see Run Control Command Group with Old Master (M4OLD) and New Master (M4NEW) Entries on page D-60.

There are three subfiles (SUBF1, SUBF2, SUBF0). SUBF1 and SUBF2 contain output data records from the application and SUBF0 contains the file definition statements describing the subfiles. The name entries of CUSTSUB, ITEMSUB, and DEFS4SUB are names you assign so you can refer to the specific subfile in the Subfile Output command group.

**Note:** Any file definitions requested for Report Summary files in an application will also be output to this subfile.

The SUBF0 has an additional specification of AUTODEF. This designates it as the subfile to receive the file definition statements describing the other subfiles in the application. Once a subfile is designated to receive the file definition statements, it will receive them for all subfiles in the application and cannot be used for any other purpose.

The following figures illustrate how to specify the content and format of the subfiles. Figure 7-5 shows the Subfile Output command group for the subfile output procedure called CUSTPART.

```
CUSTPART: EXTRACT   FILE CUSTSUB,   AUTODEF,
  COLUMNS O.CUSTNO, 1.ITEMNO, T.COUNT, F.DATE, T.OTHER, 1.ITEMNAME
```

Figure 7-5 Subfile Output Command Group

**Note:** For the expanded source listing of this example, see Run Control Command Group with Old Master (M4OLD) and New Master (M4NEW) Entries on page D-60.

In the application, this subfile output procedure (CUSTPART) is used to output a record to the subfile CUSTSUB. The Destination Name entry of CUSTSUB links you to the FILE SUBF1 entry for M4SUBF1 shown in Figure 7-4.

This subfile contains a newly created record that consists of several fields from work areas available in the application: selected fields from the old master and the coordinated file input work areas and some temporary fields created within this application. The fields are placed in the output record one after the other in the order specified, left to right, top to bottom.

If automatic file definition statements are output, the field names, without the work area qualifier, are used. Multiple references to the same field, or duplicate field names after the qualifier is dropped, result in diagnostic messages during a definition/maintenance run. To avoid this situation, set up a procedure or request that replaces the fields with duplicate names into temporary or working storage fields that have unique names. Then, you can specify the unique field names on the Extract command in the Subfile Output command group.

The field characteristics placed in the automatic file definition for the subfile are the same as the selected field, as obtained from the selected field's definition (such as an input file definition or temporary field specification). Any partial field length specification is defined as the field's length.

Because each file definition requires a key field, VISION:Builder uses the first field specified as the record key of the output subfile in the generated file definition. In the example, CUSTNUM is the record key. Note that records are written in the order that they are generated by the application, not necessarily in the order of the subfile's key.

When specifying a subfile output using selected fields, you can only create a record structure that contains one segment, the root segment. The segment (record) can be fixed or variable in length.

The EXTRACT command keywords SELECT CONTROL and ABBREVIATED control the output of fields from multiply-occurring segments. The default method is to output a record for each segment occurrence of the selected data fields. If you specify SELECT CONTROL, output occurs only once using the data from the first occurrence of the multiply-occurring segments. If you specify ABBREVIATED, output occurs only once when repeated in multiply-occurring segments. See Chapter 8, *Data Structure Handling* for more information on processing data structures.

Finally, consider the subfile format characteristics. Figure 7-6 shows the Subfile Output command group.

```
CUSTPART: EXTRACT   FILE CUSTSUB,   AUTODEF,
          RECFM VARIABLE,  BLKSIZE 1000 ,
  COLUMNS O.CUSTNO, 1.ITEMNO, T.COUNT, F.DATE, T.OTHER, 1.ITEMNAME
```

Figure 7-6 Subfile Output Command Group

**Note:** For the expanded source listing of this example, see Run Control Command Group with Old Master (M4OLD) and New Master (M4NEW) Entries on page D-60.

To enter your output subfile information, specify a VARIABLE Record Format and a Blocking Factor of 1000 (the block size for this variable length file) on tghe EXTRACT command. If you do not specify output characteristics, VISION:Builder defaults to a variable record format with a blocking factor equal to the M4REPO block size in M4PARAMS. Every time this subfile output procedure is called, the defined records are output.

Figure 7-7 shows the Subfile specifications for the subfile output procedure called ITEMSALL.

```
ITEMSALL: EXTRACT   FILE ITEMSUB     ENTIRE 1
```

Figure 7-7 Subfile Definition Showing Subfile Specifications

**Note:** For the expanded source listing of this example, see Run Control Command Group with Old Master (M4OLD) and New Master (M4NEW) Entries on page D-60.

In the application, the subfile output procedure (ITEMSALL) is used to output a record to the subfile ITEMSUB. The Destination Name entry of ITEMSUB links you to the FILE SUBF2 statement for M4SUBF2 shown in Figure 7-4.

This subfile contains a copy of the entire record from the coordinated file ITEMS (M4CORD1) work area.

The ENTIRE entry on the EXTRACT command specifies that a subfile output record is to be a copy of another input file's work area (that is, a copy of that file's record). All you need to do is specify the qualifier of the input file work area that is the source of this subfile's output (in this case, 1). Whenever this subfile output request is called, the entire record in the designated input file work area is written to the subfile. Remember, you can also output a record from the new master file work area to a subfile.

The automatic file definition statements generated for an entire record selected subfile is the same as the file definition used for the selected input file record.

When specifying an entire record selected subfile, the output record format and structure are identical to the selected input file record unless you can override this by entering the override information on the EXTRACT statement.

The previous coding examples for standard subfiles can just as easily be specified as extended subfiles. Here is how the application specifications would be changed.

The FILE SUBF1 NAME CUSTSUB and FILE SUBF2 NAME ITEMSUB statements can be removed since they are not needed for extended subfiles.

The FILE SUBF0 command are needed for the automatic subfile definition statement output.

The EXTRACT FILE commands become EXTRACT DDNAME Commands. For example:

```
CUSTPART: EXTRACT DDNAME CUSTSUB, AUTODEF,
          RECFM VARIABLE, BLKSIZE 1000,
COLUMNS O.CUSTNO, 1.ITEMNO, T.COUNT, F.DATE, T.OTHER, 1.ITEMNAME

ITEMSALL: EXTRACT DDNAME ITEMSUB ENTIRE 1
```

The CUSTSUB and ITEMSUB specifications, following the DDNAME keyword, are the ddnames of job control statements where the VISION:Builder application outputs the subfile records for those EXTRACT files. The DDNAMEs can be any user-defined name you choose.

Remember, a VISION:Builder application can specify up to 128 extended subfile outputs in addition to the 10 standard subfiles.

## Multiple Outputs to a Subfile

An output subfile can be produced using more than one EXTRACT command in the Subfile Output command group. Each EXTRACT command may refer to the same EXTRACT FILE or EXTRACT DDNAME name. This is possible as long as the physical output characteristics are the same (that is, they all are the same fixed length or are variable length). Remember that the link to a physical data set is by the FILE name or DDNAME name on the EXTRACT command, that refers to a specific M4SUBFn or user-defined DD/DTF name in the application's job control statements. You could select records from two or more input files and output them to the same subfile, effectively merging the input files.

## Using a Subfile to Update a Sequentially Coordinated File

You can use the subfile output mechanism to update sequentially coordinated files. You process your file as a sequentially coordinated input file using all record processing. This gives your application access to all records in the file. See Chapter 6, *Input File Processing* for more information on sequentially coordinated file processing.

Your application procedures or requests could change the data in the sequentially coordinated file's work area. Once the changes are complete, your application may call a subfile output procedure to output the entire sequentially coordinated file record to the subfile. The job control for your application run contains DD/DTF statements for your coordinated input file and output subfile that point to the same physical data set. This technique depends on the data set's access method. The coordinated file is processed in read mode and the subfile is processed in write mode. You must output every record read as input because you are actually rewriting the file at output time. You cannot change the length of the records.

## Subfile Output to Direct-Access Data Sets

You can output records to a subfile that is an existing direct-access data set (such as VSAM). You must output the records in ascending sequence and cannot output a record that has a key equal to an existing record key. This method is used to add records to the file.

## Subfile Output and IMS Databases

You can direct your subfile output to an IMS database. Because the output subfile is processed in write mode, this is an IMS load run (that is, a new database is being created). You can only use an entire record select subfile when output is directed to an IMS database. The input file that is selected to the subfile can be an IMS database itself or a non-IMS file format.

The name entry on the FILE SUBFn statement, which matches the FILE name entry on the EXTRACT command, must be the IMS DBD name for the database being created (see Figure 7-8). When specifying an entire record selected subfile, the

input file definition must contain segment names and sequence specifications that conform to the IMS DBD for the subfile. See Chapter 16, *Database Access Methodology* for more information on creating IMS subfiles.

```
FILE SUBF3 NAME ITEMDB   ; ITEMDB is also the DBD name of this subfile
;
ITEMSIMS: EXTRACT   FILE ITEMDB      ENTIRE O   RECFM     DLI
; Subfile extracting entire records from old master buffer
;                                    into DL/I database
```

Figure 7-8 DL/I Subfile Specification

**Note:** For the expanded source listing of this example, see Run Control Command Group with Old Master (M4OLD) and New Master (M4NEW) Entries on page D-60.

The previous coding example for a standard subfile output to an IMS database can easily be specified as an extended subfile. Here is how the extended subfile application specifications would be changed.

The FILE SUBF3 NAME ITEMDB statement can be removed since it is not needed for extended subfiles.

The EXTRACT FILE command becomes an EXTRACT DBDNAME command. For example:

```
ITEMSIMS: EXTRACT DBDNAME ITEMDB ENTIRE O DLI
```

The IMS subfile is known to IMS by the DBD name of ITEMDB.

You can also output an entire record from an IMS input file to an output subfile (that is, not an IMS database). When creating the subfile record, segment count fields are added to each parent segment if they are needed. Segment count fields are not required in IMS databases, but are required for non-IMS databases. The segment count fields that are added to the end of parent segments are four-byte fixed-binary format. When segment count fields are added, the output subfile record does not match the input file record.

The file definition of the input file does not map the output file. If you request that automatic file definition statements be generated for your output subfile, the input file definition is copied. The record format specification and characteristics are adjusted. The file definition also includes statements to define the added segment count fields. They are named in the format QsssnQQQ, where Q is a literal, the sss is a three-digit segment number, and n is the level number of the segment being counted.

Multiple subfile outputs to the same IMS subfile are not allowed.

## Subfile Output and Relational Tables

You can direct your subfile output to a relational table (DB2). There are several options available that determine the processing method used on the relational output subfile. New relational tables can be created, defined to DB2, and new rows inserted. You can delete existing rows and insert new rows into an existing relational table. You can insert rows into an existing relational table. Finally, you

can drop an existing relational table, create and define to DB2 a new version of the relational table, and then insert new rows. To perform these actions, you must have the proper DB2 authorization.

You can only output to relational table subfiles by specifying the individual fields for the columns in the rows of your relational table. Each subfile output only applies to one relational table. The relational table column name is the VISION:Builder field name specified on the COLUMNS keyword of the EXTRACT command. If you want to use column names that are different than the VISION:Builder field names, you need to supply a DB2 field name in the file definition for the selected field. DB2 field names can be specified in any file definition, even non-relational definitions. The DB2 name is only used when you specify the field as output to a relational subfile. See Chapter 3, *Defining Data* for more on defining data.

Figure 7-9 shows the specifications of a relational output subfile on the FILE SUBFn statement in the Run Control command group and on the EXTRACT statement in the Subfile Output command group.

```
CONTROL DB2 D61A INM4CALL
FILE MASTER INPUT, NAME TEMPL, KEYS NONE
;
FILE SUBF3, NAME SQLOUT, TABLE "HANME02.TESTABLE" CREATE
;
SUBFILE3: EXTRACT FILE SQLOUT, RECFM DB2,
        COLUMNS N.EMPNO N.JOB N.LASTNAME N.FIRSTNME
```

Figure 7-9 DB2 Subfile Specification

**Note:** For the expanded source listing of this example, see DB2 Subfile Specification on page D-61.

Note that the term *Create* means that the information required to define the relational table to DB2 will be generated and passed to the relational database manager.

The previous coding example for a standard subfile output to a relational table (DB2) can just as easily be specified as an extended subfile. Here is how the application specifications would be changed.

The FILE SUBF3 NAME SQLOUT statement can be removed since it is not needed for extended subfiles.

The EXTRACT FILE command becomes an EXTRACT TABLE command. For example:

```
SUBFILE3: EXTRACT TABLE "HANME02.TESTABLE" CREATE,
          COLUMNS N.EMPNO N.JOB N.LASTNAME N.FIRSTNME
```

The relational table subfile is known to DB2 by the Table Name "HANME02.TESTABLE".

You can also output an entire record from a relational table input file to a non-relational output subfile, including an IMS database. See Subfile Output and IMS Databases on page 7-10 for more information.

When creating the subfile record, segment count fields are added to each parent segment if they are needed. Segment count fields are not required in relational tables, but are required for non-relational table files and non-IMS databases. The segment count fields that are added to the end of parent segments are four-byte fixed-binary format. When segment count fields are added, the output subfile record does not exactly match the input file record. The file definition of the input file does not map the output file.

If you request that automatic file definition statements are generated for your output subfile, the input file definition is copied. The record format specification and characteristics are adjusted. The file definition also includes statements to define the added segment count fields. They are named in the format QsssnQQQ, where Q is a literal, the sss is a three-digit segment number, and n is the level number of the segment being counted.

Multiple subfile outputs to the same relational table subfile are not allowed.

## Subfile Output and Removing Dynamically Deleted Master File Segments 4

In VISION:Builder, you can delete lower level segments from the new master work area dynamically during procedure and request processing. These segments are not physically removed from the new master work area until all Main procedure and request processing is completed. This means any processing and/or output references to the new master work area still have access to the information from the dynamically deleted segments. Only at the end of the procedure and request processing cycle, when the new master work area is ready for output, are the dynamically deleted segments removed from the new master work area.

The removal of dynamically deleted segments from an entire record select subfile that is based on the new master work area is activated by specifying DELETESEGS on the FILE SUBFn statement in the Run Control command group for the output subfile. When DELETESEGS is coded for the subfile, all dynamically deleted segments are removed from the new master work area before the output of the subfile record is performed. This only occurs if the subfile is an entire record select subfile based on the new master; otherwise, this specification does not have any effect on the subfile.

It is important to understand that once the dynamically deleted segments are removed from the new master work area they are no longer available to subsequent processing in the processing cycle for the current new master record. They are available in the old master work area if an old master exists.

Because the removal of dynamically deleted segments changes the structure of the new master work area, do not output the subfile while procedure and request processing is looping on lower level segments in the new master work area. Instead, place the entire record select subfile output specifications in a subfile output request that does not contain any references to lower level segments in the new master work area. Additionally, ensure the calling procedure does not contain any references to lower level segments in the new master work area.

Dynamic segment deletion requires that a procedure or request indicate which segments are to be deleted from the new master work area. A VISION:Builder system flag must be set for the delete action to occur. Lower level segments (not the root segment) in the new master work area are marked for deletion by placing blanks or zeros (for character and numeric) into the segment key field(s) and placing 2 into the DELETE flag.

New master segments can be marked for deletion in Main (type N) procedures or requests. You can also mark new master segments for deletion during the transaction processing cycle in procedures and requests that occur after the master and transaction records are aligned or after the new master record has been updated by the current transaction record (type 2, M, 3).

Remember, the appropriately marked lower level segments are deleted from the new master work area when the new master record is ready for output at the end of the processing cycle. Using the mechanism that removes dynamically deleted segments from the new master work area that is specified as an entire record select output subfile can change the data that is normally available during the processing cycle.

## Report Summary Files

VISION:Builder allows the creation of files during the report phase that contain all the control break fields and summary values produced in a report. This file can then be post-processed in any way you want. See Chapter 9, *Reporting* for more information on specifying control breaks and summaries.

A Report Summary file can be created for selected reports in the run. This file is produced during the report phase, unlike subfiles that are produced during the processing phase. The option of generating a VISION:Builder file definition for each Report Summary file is also provided.

During the process of generating and printing report lines, VISION:Builder detects that a control break has occurred. As each control break occurs, a record is formulated and written to the Report Summary file. The record contains all control break fields and summary values for the current control break. When the report phase completes, the Report Summary files and corresponding file definitions can be made available to any application to manipulate and report the summaries as wanted. This capability is supported under OS/390, CMS, and VSE.

A Report Summary file is specified using the SUMFILE keyword on the FORMAT command of the REPORT statement (shown in Figure 7-10). Enter the name of the Report Summary file and the AUTODEF keyword if a file definition is to be generated. This name can be up to eight characters long and will be used as the DTF/DD name in the JCL for this Report Summary file, as well as the name of the generated file definition.

If a generated file definition is wanted, the AUTODEF keyword must also exist for a FILE SUBFn statement in the Run Control command group. No FILE statement is needed in the Run Control command group to specify the Report Summary file itself.

```
FILE SUBF0 NAME DEFS AUTODEF ; holds generated file definitions
                             ; for the report summary file

COMPARE: REPORT SUMMARY_ONLY, TYPE SUBROUTINE,
           COLUMNS WORKDEPT, SEX, SALARY, BONUS, COMM
      FORMAT SUMFILE SUMRPTDD, AUTODEF, HEADINGS NAME
```

Figure 7-10 Report Summary File Specification

**Note:** For the expanded source listing of this example, see Report Summary File Specification on page D-62.

Subfiles, generated file definitions, Alternate List files, and Alternate Report files can be specified simultaneously with Report Summary files without change. A single application can produce subfiles, Report Summary files, generated file definitions for these files, Alternate Report Files, and route the reports to Alternate List Files. These can be accomplished in any combination by coding the individual specifications in the Report and Subfile Output command groups.

Each Report Summary file must have corresponding JCL statements in the job stream for the appropriate DTF/DD name for the Report Summary file. For a single-step job, the JCL statements are included in the MARKIV step. For a three-step job without sample reports, the JCL statement for each Report Summary file must be included in the report (third) step. For a three-step job with sample reports, the JCL statement for each Report Summary file must be included in the processing (first) step.

If generated file definitions for the Report Summary files are wanted, there must be a corresponding JCL statement in the job stream with the appropriate DTF/DD name for the subfile that is used to capture the generated file definitions. This JCL statement is included in the first step for both single-step and three-step jobs.

For VSE, the default logical unit for all Report Summary files is SYS056. To direct output to a different logical unit, a logical unit is coded on the EXTENT statement.

The Report Summary file is opened for output at the beginning of a report and closed at the end of the report. It is always created as a variable blocked file. The block size defaults to the installation-specified block size for M4REPO. Any OS/390 system at the appropriate DFP level will create the Report Summary file with a block size determined by the system instead of the above default. You do not need to enter anything to activate this capability. You can suppress this capability by explicitly coding a block size in the JCL.

Each time a control break occurs, a record is output to the Report Summary file. In addition, a record is written at the end of the report (grand level) if AUTOGRAND is in effect for the report. This is determined from either M4PARAMS or from the specification on the Report Summaries window. A record is also written at the end of the report if any summaries are requested at level G (grand).

A report is always generated if there is a Report Summary file specified in the run. Any specification that contradicts this by attempting to suppress the report results in a type 3 message.

A No Data Selected report will generate a Report Summary file with one record containing control break fields flagged as invalid and summary values flagged as incomplete. If the No Data Selected report is suppressed, the Report Summary file has no records written to it and remains empty.

A Report Summary file cannot be requested in a run where report optimization is specified using the SUMMARIZE keyword on the CONTROL command in the Run Control command group. In a three-step run, a type 6 message is issued in the report step and the run terminates. In a single-step run or in a three-step run with sample reports, a type 3 message is issued at compile time for the report and the associated request is invalidated.

Access to Report Summary files through User I/O or dynamic own-code hooks is not provided. However, in an application that involves static own-code, your M4OWN routine is responsible for the I/O of all files. This includes the Report Summary file(s), if requested. At Hook 21 time (each PUT I/O), the secondary calling list contains a file code for the current file for which the output is to be performed. In the case of a PUT to a Report Summary file, the file code is set to X'30'. Refer to the *VISION:Builder for OS/390 Environment Guide* (OS/390 users) or the *VISION:Builder for VSE Environment Guide* (VSE users) for a complete description of own-code hooks.

If a Program Analyzer Cross Reference is requested in a run that produces Report Summary files, the related file information for the Report Summary file(s) is included in the File Cross Reference records of the Program Analyzer subfile. The Program Analyzer is not supported by VISION:Workbench for DOS. See Chapter 20, *Program Analyzer* for more information on the Program Analyzer.

## Record Format

The Report Summary file record consists of a header followed by each control break field and summary requested in the report. The control break fields and summaries are placed into the record in the order they appear in the field output specifications. Each of these is followed by a one-byte status field indicating the validity and existence of the element.

The header contains information that identifies where a record originated: the request name and report number (1-9). A record number computed by VISION:Builder serves as the record key and is incremented each time a record is written to the file. The control break level at which the record is written is also included in the header.

The following lists the type, length, and scale guidelines:

- If the element is a control break field, the field in the Report Summary file record is of the same type, length, and scale as the control break field.
- If the element is a TOTAL, CUM, or AVERAGE summary, the field in the Report Summary file record is a packed field with a length of 15 and a scale equal to that of the field being summarized.
- If the element is a PERCENT or RATIO summary, the field in the Report Summary file record is a packed field with a length of 15 and a scale equal to that specified for printing of the PERCENT or RATIO.
- If the element is a COUNT summary, the field in the Report Summary file record is a fixed field with a length of 4 and a scale of zero.
- If the element is a MAX or MIN summary, the field in the Report Summary file record is of the same type, length, and scale as the field being summarized.

The Report Summary file contains a status indicator for each control break field and summary element in the record. The purpose of this one-byte character field is to indicate whether a field is valid, incomplete, invalid, or not available at a given control break level. See the following table.

| Status | Value | Control Break Field | Summary Element |
|---|---|---|---|
| Valid | 0 | Yes | Yes |
| Incomplete | 1 | No | Yes |
| Invalid | 2 | Yes | Yes |
| Not Available | 3 | Yes | Yes |
| Contents of a field are set to binary zeroes whenever the status is invalid or not available. | | | |

At the time a control break occurs, VISION:Builder determines whether or not an element is available at the current control break level. If it is, the status is set to valid, incomplete, or an appropriate value is inserted into the record. Otherwise, it is set to binary zeros and the status indicator is set to indicate not available.

### Not Available

The status of a summary is not available when its associated control break level is more major than the current control break level. When this occurs, a value of binary zeros is assigned to the field in the Report Summary file record. For example, the total of field SALARY at level 2 is assigned binary zeros for any control break 3 through 9. At the grand control break level, all summaries are available and are included in the record.

The status of a control break field is not available when its associated control break level is more minor than the current control break level. The status can also be not available if the control break field is a missing field. These fields print as a dash (-) on a report. The dash is an installation option. When a control break field has a status of not available, a value of binary zeros is assigned to the field in the Report Summary file record and the status indicator is set to indicate not available. For example, a level 3 control break field is assigned binary zeros for any control break 0 (grand) through 2. At the grand control break level, all control break fields are set to binary zeros and flagged as not available.

### Invalid

The status of a control break field or a summary is invalid if, according to VISION:Builder, it is an invalid field. An invalid field or summary prints as an asterisk (*) on a report. The asterisk is an installation option. In addition, a summary element is invalid if it contains a value that cannot be contained in a packed field with a length of 15. When this occurs, a value of binary zeros is assigned to the field in the Report Summary file record. See Chapter 4, *Field Characteristics and Handling* for the complete definition of an invalid field.

### Incomplete

An incomplete status applies only to a summary element. A summary is valid, but has an incomplete status when one or more missing or invalid fields participated in computing the summary. Incomplete summaries can be optionally flagged on a report with a user-specified character. When this occurs, the summary value as computed by VISION:Builder is inserted into the record.

### Valid

If a control break field or summary has not been classified as invalid, incomplete, or not available according to the above rules, its status is valid and its contents are inserted into the record.

## Generated File Definition

A file definition that maps each Report Summary file record can be optionally generated and output to a subfile. The generation of a file definition for a Report Summary file is specified by adding the AUTODEF keyword to the REPORT statement, as shown in Figure 7-10 on page 7-15. The subfile that contains the generated file definition is specified by coding AUTODEF on a FILE SUBFn statement. If definitions are requested but this subfile has not been specified, a type 3 message is issued. Also note that only one such subfile can be specified. Generated file definitions for all Report Summary files and standard subfiles are written to this file in the order they appear in the application.

Note that file definitions for standard subfiles are also generated by specifying a subfile that contains the generated definitions (AUTODEF on a FILE SUBFn statement). The AUTODEF keyword may only be specified on one FILE SUBFn

statement., Therefore, all generated file definitions, whether they are from standard subfiles or Report Summary files, are written to the same definition subfile.

As each individual report is compiled, the file definition for the Report Summary file is generated and written out to the designated subfile. The name of the file definition for the Report Summary file is the same name entered for the Report Summary file in the FORMAT command on the REPORT statement.

The file definition statement contains a record format of variable and a buffer size equal to the installation-specified block size for M4REPO. This is true even if the operating system is allowed to determine the actual block size for the file. If system-determined block size is used, VISION:Builder can adjust for any discrepancy when the file is read as an input file.

The statements generated for the fields in the record header are a fixed set. They define the fields and column headings for the fields in the record header. The record header is 16-bytes long.

VISION:Builder constructs an individual statement set (a set of Ln statements - L0, L1, L2, and L3) for each control break field and its status indicator and each summary element and its status indicator. The statement sets are generated in the order that the fields appear in the COLUMNS command of the REPORT statement, one for each type of summary per field. The starting location of the first field following the header record is byte 17.

Each control break field and summary element is assigned a sequential number (nnnnn), starting with 1, ordered according to the sequence of field output specifications. Each summary type requested for a field is assigned a separate number. The number indicates that it is the nth field in the summary record and is used in generating unique field names for the statement set (Ln statements). nnnnn is appended to a three character prefix to formulate a field name. Leading zeros in nnnnn are represented with an underscore (_) to improve readability.

The field name for a control break field is GRPnnnnn. The type, length, and scale are derived from the definition of the field on which the control break is to occur.

The field name for a summary element is a three-byte prefix followed by nnnnn for the field. The three-byte prefix corresponds to the summary type:  TOT, CUM, CNT, MAX, MIN, AVG, PCT, and RTO. For example, TOT___1 would be the field name if the first Report Detail statement requested a TOTAL. The type, length, and scale are completed according to the guidelines listed on the preceding pages.

Because the scale of a summary on a character field is not known until report time, the scale is left blank and a type 1 warning message is issued at the time the file definition is generated. This alerts you that the decimal places can need adjusting.

The field name for a control break or summary element's status indicator is STSnnnnn. It is a one-byte character field.

The TOTAL, CUM, or AVERAGE of a time processing field (type L or S) is a packed field with a length of 15 and held as the total number of seconds. When it is reported directly from the Report Summary file, it does not print in the

hours:minutes:seconds format. To accomplish this, it must first be moved to a field defined as L or S to prompt the conversion. Subsequent output from this converted field will yield the wanted format.

Figure 7-11 shows the report detail for a report that requests a Report Summary file.

```
COMPARE: REPORT SUMMARY_ONLY, TYPE SUBROUTINE,
           COLUMNS WORKDEPT, SEX, SALARY, BONUS, COMM
      FORMAT SUMFILE SUMRPTDD, AUTODEF, HEADINGS NAME
       ; report is captured in summary report file
       ;        identified with DD SUMRPTDD
       ; a file definition for that report file is generated
       ;        and captured in the definition subfile SUBF0
       ;        identified with DD M4SUBF0
       ;
       ORDER BY WORKDEPT, SEX
        GROUP AT LEVEL 2 BY WORKDEPT, SEX
         AVERAGE SALARY BY SEX
         AVERAGE BONUS  BY SEX
         AVERAGE COMM   BY SEX
         MAX SALARY BY SEX
         MAX BONUS  BY SEX
         MAX COMM   BY SEX
         COUNT BONUS BY WORKDEPT
    TITLE 'COMPENSATION COMPARISON BY DEPARTMENT AND GENDER'
END REPORT
```

Figure 7-11 Report Detail for a Report that Requests a Report Summary File

**Note:** For the expanded source listing of this example, see Report Summary File Specification on page D-62.

Figure 7-12 shows the generated file definition from this same report. The column headings that are generated are terminated with the system delimiter designated by the user for the application. For the purposes of this example, a pound sign (#) is used as the system delimiter.

```
ASLRC   RC                  #
SUMRPTDDFD          V        4088
SUMRPTDDL0HRECNUM  011  01 04F1
SUMRPTDDL1HRECNUM                          RECORD#
SUMRPTDDL2HRECNUM                          NUMBER#
SUMRPTDDL0HREQNAME 011  05 08C
SUMRPTDDL1HREQNAME                         REQUEST#
SUMRPTDDL2HREQNAME                         NAME#
SUMRPTDDL0HRPTNUM  011  13 02F
SUMRPTDDL1HRPTNUM                          REPORT#
SUMRPTDDL2HRPTNUM                          NUMBER#
SUMRPTDDL0HSUMLEV  011  15 01C
SUMRPTDDL1HSUMLEV                          SUMMARY#
SUMRPTDDL2HSUMLEV                          LEVEL#
SUMRPTDDL0GRP___1 011  17  3C
SUMRPTDDL1GRP____1                         WORKDEPT#
SUMRPTDDL2GRP____1                         GROUP#
SUMRPTDDL0STS____1 011  20 01C
SUMRPTDDL1STS____1                         WORKDEPT#
SUMRPTDDL2STS____1                         GROUP#
SUMRPTDDL3STS____1                         STATUS#
SUMRPTDDL0GRP____2 011  21  1C
SUMRPTDDL1GRP____2                         SEX#
SUMRPTDDL2GRP____2                         GROUP#
SUMRPTDDL0STS____2 011  22 01C
```

Figure 7-12 Traditional Syntax for Report Summary File Specifications (Page 1 of 2)

```
SUMRPTDDL1STS___2                                SEX#
SUMRPTDDL2STS___2                                GROUP#
SUMRPTDDL3STS___2                                STATUS#
SUMRPTDDL0MAX___3 011  23  5P  2
SUMRPTDDL1MAX___3                                SALARY#
SUMRPTDDL2MAX___3                                MAX#
SUMRPTDDL0STS___3 011  28 01C
SUMRPTDDL1STS___3                                SALARY#
SUMRPTDDL2STS___3                                MAX#
SUMRPTDDL3STS___3                                STATUS#
SUMRPTDDL0AVG___4 011  29 15P  2
SUMRPTDDL1AVG___4                                SALARY#
SUMRPTDDL2AVG___4                                AVERAGE#
SUMRPTDDL0STS___4 011  44 01C
SUMRPTDDL1STS___4                                SALARY#
SUMRPTDDL2STS___4                                AVERAGE#
SUMRPTDDL3STS___4                                STATUS#
SUMRPTDDL0CNT___5 011  45  4F
SUMRPTDDL1CNT___5                                BONUS#
SUMRPTDDL2CNT___5                                COUNT#
SUMRPTDDL0STS___5 011  49 01C
SUMRPTDDL1STS___5                                BONUS#
SUMRPTDDL2STS___5                                COUNT#
SUMRPTDDL3STS___5                                STATUS#
SUMRPTDDL0MAX___6 011  50  5P  2
SUMRPTDDL1MAX___6                                BONUS#
SUMRPTDDL2MAX___6                                MAX#
SUMRPTDDL0STS___6 011  55 01C
SUMRPTDDL1STS___6                                BONUS#
SUMRPTDDL2STS___6                                MAX#
SUMRPTDDL3STS___6                                STATUS#
SUMRPTDDL0AVG___7 011  56 15P  2
SUMRPTDDL1AVG___7                                BONUS#
SUMRPTDDL2AVG___7                                AVERAGE#
SUMRPTDDL0STS___7 011  71 01C
SUMRPTDDL1STS___7                                BONUS#
SUMRPTDDL2STS___7                                AVERAGE#
SUMRPTDDL3STS___7                                STATUS#
SUMRPTDDL0MAX___8 011  72  5P  2
SUMRPTDDL1MAX___8                                COMM#
SUMRPTDDL2MAX___8                                MAX#
SUMRPTDDL0STS___8 011  77 01C
SUMRPTDDL1STS___8                                COMM#
SUMRPTDDL2STS___8                                MAX#
SUMRPTDDL3STS___8                                STATUS#
SUMRPTDDL0AVG___9 011  78 15P  2
SUMRPTDDL1AVG___9                                COMM#
SUMRPTDDL2AVG___9                                AVERAGE#
SUMRPTDDL0STS___9 011  93 01C
SUMRPTDDL1STS___9                                COMM#
SUMRPTDDL2STS___9                                AVERAGE#
SUMRPTDDL3STS___9                                STATUS#
```

Figure 7-12 Traditional Syntax for Report Summary File Specifications (Page 2 of 2)

**Note:** For the expanded source listing of this example, see Report Summary File Specification on page D-62.

## Rejected Transactions File❹

In an application run that is using a transaction input file, any transaction records that are rejected can be directed to an output file. To specify that you want to output the rejected transaction records, include a FILE REJECT command in the Run Control command group and this file will be automatically handled by VISION:Builder.

All opens, closes, and writes are performed automatically. The format and structure of the output file is identical to the defined values of the input transaction file. The records are all the rejected transactions from the application. This includes the records that were rejected by the VISION:Builder system and the records that were procedurally rejected by the application.

## Deleted Master File Records❹

In an application run that is deleting master file records, you can create an output file that contains a copy of all the master records that were deleted. To specify that you want to output the deleted master file records to a separate file, include a FILE AUDIT command in the Run Control command group and the file will be automatically handled by VISION:Builder.

All opens, closes, and writes are performed automatically. The format and structure of the output file is identical to the defined values of the master file. This option is not intended for use with IMS database master files. The M4AUDIT file is given a RECFM=U (for undefined) when used with an IMS master file. Undefined records are limited to a record size of 32,760. This option cannot be used with relational master files.

## OS/390 System-Determined Block Size Support

An OS/390 operating system with DFP version 3 Release 1.1 or higher can dynamically determine the optimum block size for output files that are assigned to tape or disk under certain conditions.

VISION:Builder automatically determines if the operating system is OS/390 and at the required DFP level. If so, all output files in the run meeting the OS/390 requirements are created with a block size determined by the operating system.

Output files potentially involved are M4NEW,❹ M4SUBFn, M4LIST, M4REPO, M4PAOUT, M4REPn, M4AUDIT,❹ M4REJCT,❹ Alternate M4LISTs, and Report Summary files.

The blocking factor and buffer size specified in the file definition or on the Subfile EXTRACT command for a file have no effect on the block size. It serves only as an indicator that the file is blocked. A blocking factor of 1 indicates that the file is unblocked.

The default is to let the OS/390 operating system determine the block size, ignoring the block size and blocking factor specified in the application. To override this default and prevent the operating system from determining the block size, specify a non-zero BLKSIZE on the DD statement for the output file. Note that this overrides the value obtained from an input file definition when the input file is entire record selected to an output subfile. The output file is created with the block size on the DD statement.

Subfiles with a fixed record format that do not specify a subfile blocking factor are created as fixed, unblocked if the operating system *does not allow* system-determined block size. If the operating system *does allow* system-determined block size, the resulting file format is fixed, blocked with the block size being determined by the operating system. To create a fixed, unblocked subfile under a system that supports system-determined block size, either code RECFM=F on the DD statement or specify BLKSIZE 1 on the EXTRACT command in the Subfile Output command group.

When the system supports system-determined block size, VISION:Builder does not allocate the buffers for any input or output files in the run. They are allocated by the operating system. This allows you to concatenate, in any block size order, sequential data sets on DASD that are accessed by QSAM and use OS/390 system-allocated buffers. As a result of this, the processing phase of VISION:Builder can abend with an out of storage condition when processing concatenated data sets.

Because VISION:Builder cannot predetermine the maximum block size for the files, there is the potential for insufficient storage for the buffer each time the next data set in the concatenation sequence is opened.

Specifics on system-allocated buffers, how the block size is determined, and what conditions must be satisfied in order for the system to determine the block size for a file are documented in IBM's OS/390 *Data Administration Guide*.

## Source and Report Listings File

All the printed output produced by VISION:Builder is directed to a list file known as M4LIST. The output list file or data set characteristics are sequential fixed length records. The record length is normally 133, which is the standard M4LIST width setting in the M4PARAMS module plus 1 for the ASA control character. If the standard M4LIST width value in the M4PARAMS module is increased, the M4LIST record size increases. Refer to the *VISION:Builder for OS/390 Installation Guide* (OS/390 users) or the *VISION:Builder for VSE Installation Manual* (VSE users) for details on the M4PARAMS module.

In application processing runs, the source statement listing, messages, statistics, and the application generated reports are directed to M4LIST. In definition/maintenance runs, the source statement listing, messages, and the

definition glossaries are directed to M4LIST. In the common library utility runs, the source statement listing, messages, and the formatted output of the utility operation are directed to M4LIST.

# Alternate List File for Report Listings

In VISION:Builder application processing runs, the application can specify an alternate output destination for all the application generated reports. This allows you to separate the application generated reports from the source listing, messages, and statistics. This is known as Alternate List Facility.

You can direct all the reports to a single alternate list destination. You can direct each individual report to a separate alternate list destination. Plus, you can direct several reports to the same alternate list destination. You can specify up to 255 separate alternate list destinations. The DTF/DD name for your alternate list destinations is a unique name specified in your report output definition. The default alternate list DTF/DD name is M4LIST1 when no name is specified.

To specify alternate list destinations for your report output, you specify METHOD ALTLIST and enter the DTF/DD name of the alternate list file as the DDNAME on the FORMAT command in the Report command group. Figure 7-13 shows you an example.

```
DEPTSUM: REPORT SUMMARY_ONLY, TYPE SUBROUTINE,
           COLUMNS WORKDEPT, SALARY, BONUS, COMM, T.COMP
      FORMAT HEADINGS NAME, METHOD ALTLIST DDNAME DEPTTOTS
```

Figure 7-13 Report Definition Showing the Alternate List File Entry

**Note:** For the expanded source listing of this exampleCOLUMNS, see Report Summary File Specification on page D-62.

Note that the DEPTSUM report is being directed to the DTF/DD named DEPTTOTS.

To direct all application reports to the alternate list file, specify ALTLIST YES on the LISTCNTL command in the Run Control command group.

To accomplish this using fixed format syntax, use the Report Handling and DTF/DD entries in the Output Specifications (En) statement.

The Source Statement Listing entry in the fixed format Run Control (RC) statement can be used to direct all the application reports to the alternate list file DTF/DD named M4LIST1.

Remember, you must supply the appropriate job control statements for the specified DTF/DD names so that the application processing run can perform the output processing.

# Alternate Report Output Methods

In addition to the Report Listings File and Alternate List File described above, VISION:Builder provides five additional methods which can be used to output the content of a report. A method in this context refers to how the data content for the report is prepared for output. The term report in this context refers to the content of the output as specified using the usual report specification statements. However, when these alternate report output methods are specified, the data that would typically be prepared for a report page layout is instead prepared in one of the five special formats that are more suitable for use by other software tools.

The output for these report methods is always directed to a unique ddname (SYSnnn in VSE) specified in the JCL. Depending upon the method used, certain specifications intended specifically for a printed page layout are syntactically accepted but functionally ignored. This is done to make it easy to change the report statement specifications from a form suitable for a printed page to an alternate output method and vice versa.

VISION:Builder supports the following five alternate report output methods. The table below lists the specifications for the ASL METHOD keyword value used on a FORMAT command, as well as the fixed format En statement. See the *VISION:Builder ASL Reference Guide* or *VISION:Builder Reference Summary* for further information.

| Alternate Report Output Method | Specification for | | Description |
|---|---|---|---|
| | En Statement | ASL METHOD | |
| Comma-Delimited Data (.csv) | CD | CSV | Output data fields as a comma-delimited file |
| Tab-Delimited Data (.txt) | TD | TAB | Output data fields as a tab-delimited file |
| Plain Text File (.txt) | PT | PLAINTEXT | Output report as a plain text file |
| HTML Document (.html) | HT | HTML | Output report as an HTML document |
| Raw Data Output | RD | RAWDATA | Output report as a variable length data File. |
| | RO | RAWDATA with AUTODEF | Output report as a variable length data File and generate a file definition. |

## Comma-Delimited Data (CSV)

The Comma-Delimited Data method (sometimes known as Comma-Separated Variables) is used for preparing data for import into spreadsheets or other tools that will accept this form of input. It formats the columns of each detail or summary line as a series of fields delimited by commas. This output is restricted to detail only (no summaries) or summary only (no detail) data. Column heading lines are also formatted as delimited data lines, unless column headings were suppressed. No preface lines or page title lines appear on the output. (If you specify either preface lines or page title lines, they are ignored.)

If the data for any column contains a comma, the column contents are bracketed by quotation marks (“). If the data for any column contains a quotation mark (“), the quotation mark are changed to an apostrophe (‘).

For comma-delimited data output for your report, specify FORMAT METHOD CSV and the DDNAME (SYSnnn in VSE) in the Report command group. In Fixed Format syntax, use the Output Specifications (En) statement to specify Report Handling and ddname (SYSnnn for VSE).

## Tab-Delimited Data

The Tab-Delimited Data method is used for preparing data for import into spreadsheets or other tools that will accept this form of input. It is nearly identical to the Comma-Delimited Data method except that the columns of data are delimited by a tab character rather than a comma. If the data for any column contains a tab, the character representing the tab will be changed to a blank. No quotation marks, to bracket the column contents, are ever inserted.

For tab-delimited data output for your report, specify FORMAT METHOD TAB and the DDNAME (SYSnn in VSE) in the Report command group. In Fixed Format syntax, use the Output Specifications (En) statement to specify Report Handling and ddname (SYSnnn for VSE).

## Plain Text File

The Plain Text method is used for preparing data for insertion into other text documents. It prepares the report content without machine or platform-specific spacing control characters preceding each line. Blank lines are embedded into the output to simulate the line spacing that would normally appear on a printed report. A plain text report does not contain any page breaks. The page title and column headings appear once at the beginning of the report and only at subsequent P-type subtitle control breaks.

For plain text file output for your report, specify FORMAT METHOD PLAINTEXT and the DDNAME (SYSnn in VSE) in the Report command group. In Fixed Format syntax, use the Output Specifications (En) statement to specify Report Handling and ddname (SYSnnn for VSE).

## HTML Document

The HTML method is used for preparing data for viewing by many users via a corporate intranet or worldwide via the Internet. It prepares the report content for viewing with a web browser. The HTML document uses the frames feature of modern browsers to render the report data suitable for viewing and scrolling. This method requires a template containing HTML frame specifications in the M4HTBASE library (partitioned data set). A default template is delivered with VISION:Builder. You can customize the default template and create up to 99 additional variations know as styles. You can accept the default style or specify any of the available user-created styles for each HTML report. Refer to the *VISION:Builder for OS/390 Environment Guide* (OS/390 users) or the *VISION:Builder for VSE Environment Guide* (VSE users) for information regarding the creation and customization of HTML styles for VISION:Builder.

The HTML method places the report output into a partitioned data set. Members of this data set contain the HTML specifications and data for the report document. The member names in this data set are always MAIN, HEAD, BODY, or LTOC. The LTOC member is present when any level 1 subtitle specifications are present in the report definition; otherwise, it is absent. These members must be transferred, using a suitable file transfer tool, to a server or local workstation where they can be accessed by the web browser. The browser session used to view the VISION:Builder HTML document must initiate the viewing via the MAIN file (member).

For HTML Document output for your report, specify FORMAT METHOD HTML and the DDNAME (SYSnn in VSE) in the Report command group. This destination must be a partitioned data set. Additionally, a user-defined HTML style can be specified using the STYLE keyword (see the *VISION:Builder ASL Reference Guide*). In Fixed Format syntax, use the Output Specifications (En) statement to specify Report Handling, ddname, and HTML Style. The HTML Document method is only available on the OS/390 operating platforms.

## Raw Data Output

The Raw Data method is used to output the report data fields to a standard variable length file. The data fields are output in their original format. The output is similar to the processing phase subfile output with a format of V. The difference is that the file is created during the report generation phase and is ordered according to any report sequencing (sorting) specifications. The records are always variable length (RECFM=V). The output is written to the data set specified by the DDNAME keyword and value.

Chapter

# 8 Data Structure Handling

This chapter describes how the data structures defined in Chapter 3, *Defining Data* are processed by VISION:Builder both automatically and under controlled conditions.

It is important to understand that the power of VISION:Builder lies in its method of processing data structures automatically through looping. Once the essential concepts of automatic processing and controlled processing are clear, application development becomes easy.

Automatic Looping on page 8-1 discusses the various types of automatic processing or looping. The following sections discuss several ways in which you can control the automatic looping to meet the specific needs of your application.

## Automatic Looping

Suppose you have a structured file that can be represented by the hierarchical structure shown in Figure 8-1.

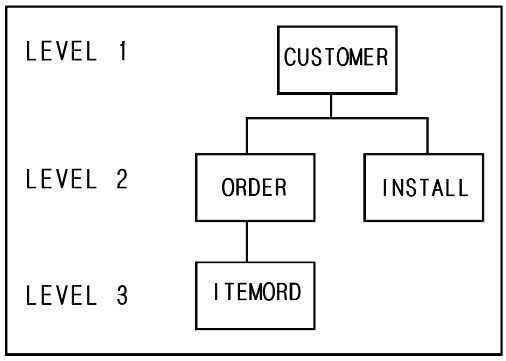



Figure 8-1 Structure of the Customer File

The CUSTOMER segment contains customer information such as name and telephone number. The ORDER segment contains order information for each customer. The ITEMORD segment contains item information for each order. The INSTALL segment contains installation information for each customer.

Each structured customer record in the customer file will have its own set of segment occurrences that contain all the related information for the specific customer. Suppose you have a customer record for customer A1 that has information for three orders: B1, B2, B3. Order B1 has two items ordered (C1, C2) and order B3 has one item ordered (C3). There is also information on one installation (D1). Figure 8-2 shows how this information is related in the structure as a set of segment occurrences.

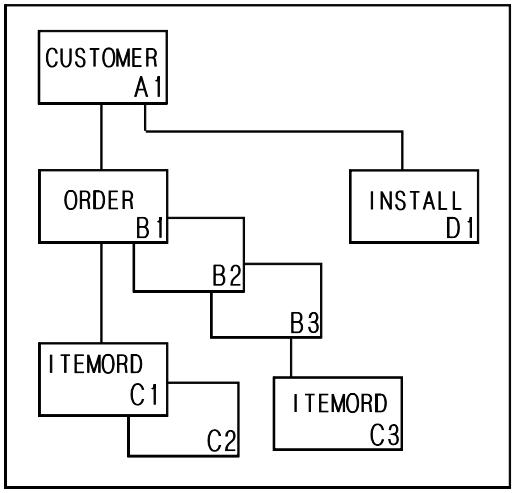



Figure 8-2 Customer A1 and Related Information

To process such structured records, VISION:Builder uses the concept of looping. Each record is presented to a VISION:Builder procedure. The first executed reference within that procedure to any field in a lower level segment starts a loop on the occurrences of that segment type. The loop is automatically generated and controlled by VISION:Builder. Each occurrence for that segment type is automatically available for further processing.

Loops are automatically controlled and processed to completion within a procedure, request, report, or subfile. Branching locations within a procedure do not alter the looping process. For example, a CONTINUE branch causes processing to cease for the current occurrence of a segment in a loop and to resume (at the first reference to that segment type in the procedure) with the next occurrence of that segment type.

When using automatic looping through lower level segments, give careful consideration to where you first reference a lower level segment.

Suppose you have two reports using the structure shown in Figure 8-2. You want to create REPORT1 (CUSTNO) and REPORT2 (ORDERNO), sorted by the CUSTNO field in the first report, and sorted by the ORDERNO field in the second report.

The following lists the specifications on the Report Detail window for the Field Name and Sort Sequence entries:

REPORT1

```
CUSTNO  1
ORDERNO
```

REPORT2

```
ORDERNO 1
CUSTNO
```

The CALL statements to call the subroutine procedure are as follows:

```
CALL REPORT1
CALL REPORT2
```

VISION:Builder loops through all occurrences of ORDERNO in REPORT1, returns to the calling procedure, and loops through all occurrences of ORDERNO in REPORT2. For CUSTNO A1, this results in all three occurrences of ORDERNO being accessed twice.

However, if you start the loop on ORDERNO before you invoke the reports, you loop only once for each occurrence. This is easily done by any reference to a field in the ORDER segment.

For example:

```
LET ORDERNO = ORDERNO
CALL REPORT1
CALL REPORT2
```

In this example, the loop on ORDERNO is performed once.

Note that if you have a large database with numerous occurrences of the lower level segment, starting the loop in the calling procedure is very efficient.

The following looping rules apply:

- The first executed reference to any field in a lower level segment starts a loop on the occurrences of that segment type.
- The loop is executed for each segment occurrence of that segment type. Each occurrence is available to the procedure for further processing.
- The range of the loop extends to the end of the procedure if begun in a procedure statement, to the end of a report if begun in a report, or to the end of the subfile definition if begun in a subfile definition.
- Nested loops (loops within loops) can occur by referencing different segment types.
- The innermost loop is executed first, the next outermost loop is executed next, and so on to the outermost loop.
- Once a lower level segment type (or array element) is referenced (thereby starting a loop), it does not start another loop if referenced again in the same procedure, report, or subfile. It is already in an automatic loop.
- Subroutine procedures (including called reports and subfiles) do not start loops on segment types already part of a loop in the calling procedure.

There are three types of loops:

- Single-path nested loops. Initiated by referencing one or more segment types along a single-path (or leg) of the hierarchical structure. See Chapter 3, *Defining Data* for a definition of a path or leg of a structure. The innermost loop is executed first, the next outermost is executed next, and so on to the outermost loop.
- Multi-path nested loops. Initiated by referencing multiple segment types along two or more legs of one or more hierarchical structures. The innermost loop is executed first, the next outermost is executed next, and so on to the outermost loop.
- Multi-path parallel loops. Initiated by referencing multiple segment types along two or more legs of one or more hierarchical structures. The loops on all paths are advanced simultaneously (in parallel). This is allowed only in reports and subfiles.

Each of these is described in detail in the remainder of this section.

## Single-Path Nested Loops

Figure 8-3 shows the simplest form of a single-path nested loop in which there is only one loop on one segment. The only loop is an outer loop on the ORDER segment, resulting in no nesting of loops.

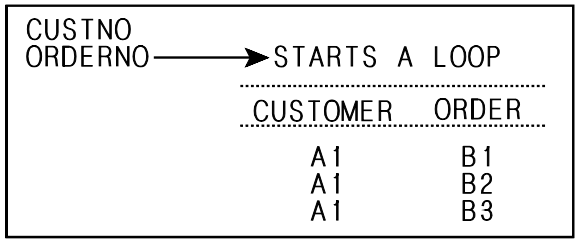



Figure 8-3 Simple Single-Path Nested Loop

Notice how the first reference to the ORDER segment starts a loop on all occurrences of the ORDER segment for customer A1. Note how VISION:Builder prints the report records, repeating A1 for each lower level occurrence.

Figure 8-4 shows an example of a simple nested loop in which more than one segment type is referenced. Two loops are initiated in this case on the ORDER and ITEM segments. These loops are nested.

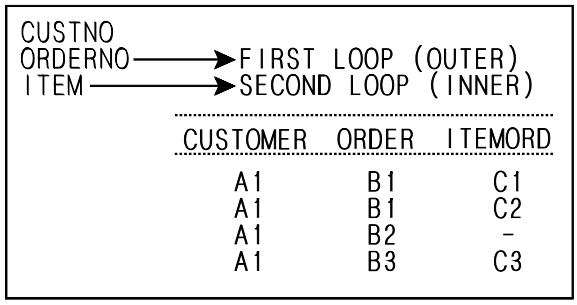



Figure 8-4 Single-Path Nested Loop

The report statement that references the ORDERNO field starts the loop. The report statement that references the ITEMORD field starts the second loop nested within the first loop. The dash prints when there are no segment occurrences.

Note how the inner loop gets executed completely before the outer loop is executed. In this example, all occurrences of the ITEMORD segment (C1 and C2) print for the first occurrence of the ORDER segment (B1) before proceeding to the second occurrence of the ORDER segment.

The order in which the lower level segment fields are referenced on the Report statement makes a difference as to the appearance of the report. If you reference the lower level segment types in an inverse order (ITEMORD first and ORDER second), the outer loop is on the ITEMORD segment and the inner on ORDER (see Figure 8-5).

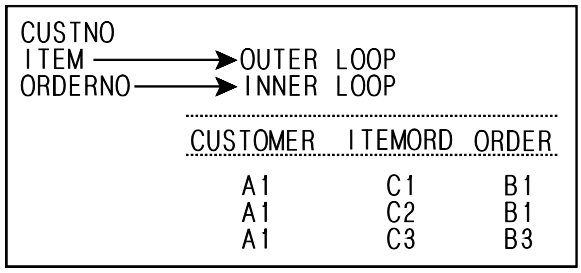



Figure 8-5 Single-Path Nested Loop in Inverse Order

Note how this report is different from the report in Figure 8-4. The ORDER segment (B2) is absent from this report. This is because the ITEMORD segment is the outer loop and controls references to the inner loop. Orders without items do not print.

Figure 8-6 shows a more complex single-path nested loop initiated by field references within a procedure. Notice how the nested loops are generated from the innermost segment type (last referenced) to the outermost segment type referenced.

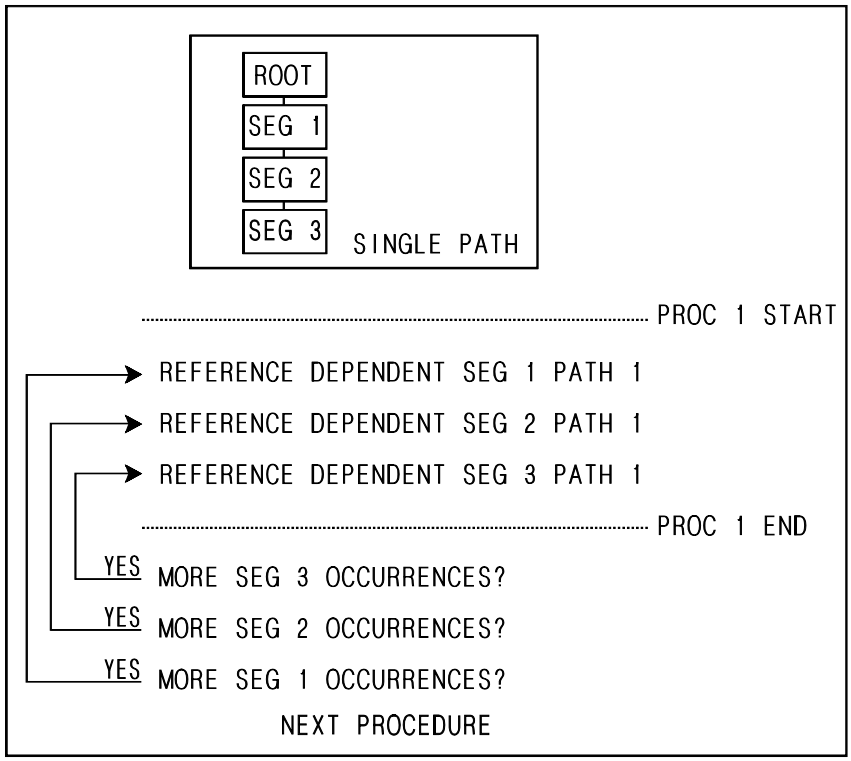



Figure 8-6 Single-Path Nested Loop in Procedures

Single-path nested looping is the default method of looping in a procedure, request, report, or subfile. You do not make a specification to activate this type of looping. It is activated automatically by VISION:Builder when you reference one or more segment types along the same hierarchical leg.

## Multi-Path Nested Loops

A multi-path nested loop is a nested loop generated by referencing multiple segment types along two or more legs of one or more hierarchical structures. This type of looping can occur in a procedure, request, report, or subfile.

For example, in Figure 8-7, the report statements that reference the ORDERNO field and the INSTNO field start nested loops along two legs of the structure

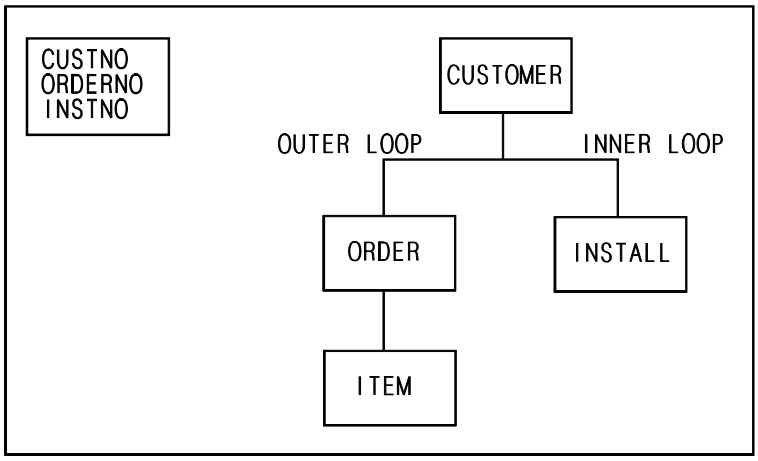



Figure 8-7 Multi-Path Nested Loops

Note that under this definition, multi-path nested loops could also occur when references are made to fields in lower level segments in the old and new work areas or when references are made to master file and coordinated file fields in lower level segments. Procedure, request, report, or subfile definition statements can initiate multi-path nested loops.

Figure 8-8 shows the flow of control in the execution of multi-path nested loops. Note that the first statement that initiates a loop is the outermost loop.

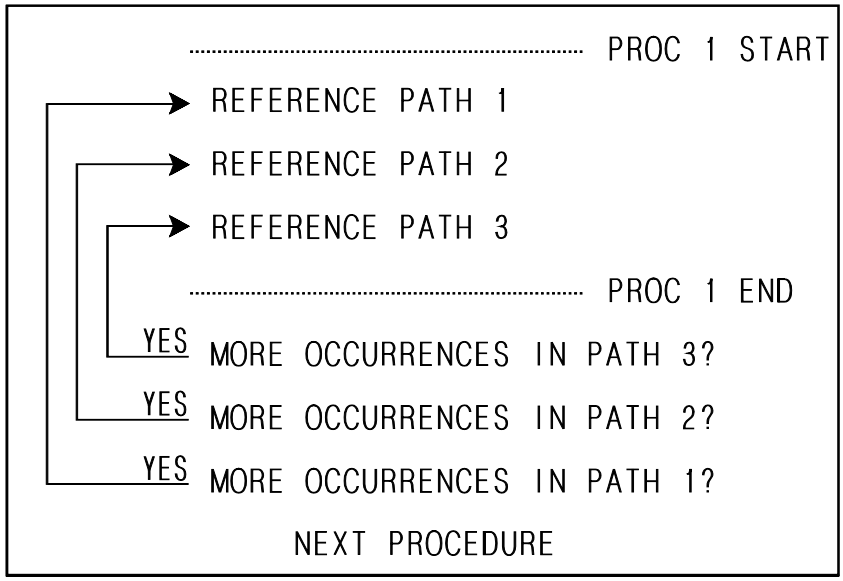



Figure 8-8 Multi-Path Nested Loops Flow of Control

All occurrences in path 3 are processed before moving to the next occurrence in path 2. At this point, the loop is begun again on path 3. Following this progression, you loop through all occurrences in path 3 repeatedly – once for each occurrence in path 2. For this reason, avoid multi-path nested looping unless you want all possible combinations of segment occurrences processed between the multiple paths.

This is illustrated by the structure in Figure 8-9

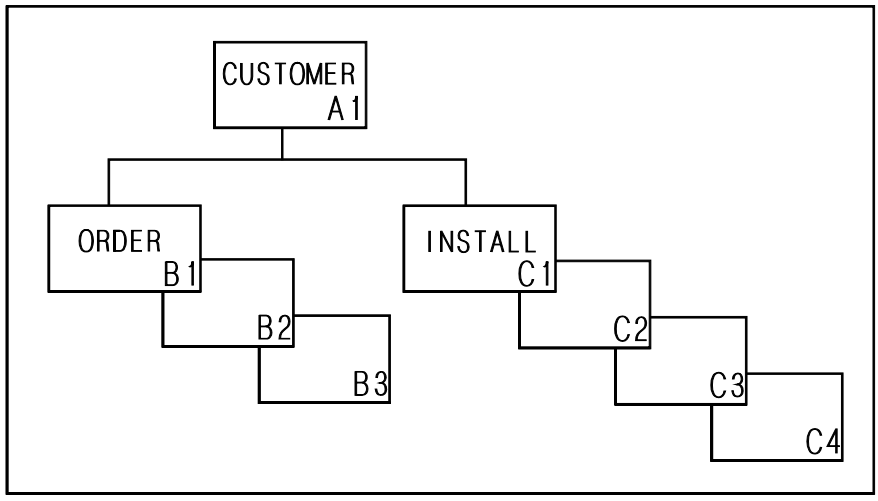



Figure 8-9 File Structures and Multi-Path Nested Looping

If you reference fields in the ORDER or INSTALL segments in a report, VISION:Builder will start a loop on the ORDER segment, then on the INSTALL segment. This is a multi-path nested loop because the ORDER segment is in a different path than the INSTALL segment.

The report is shown in Figure 8-10 with all occurrences of the INSTALL segment (the inner loop) repeated for each occurrence of the ORDER segment (the outer loop).

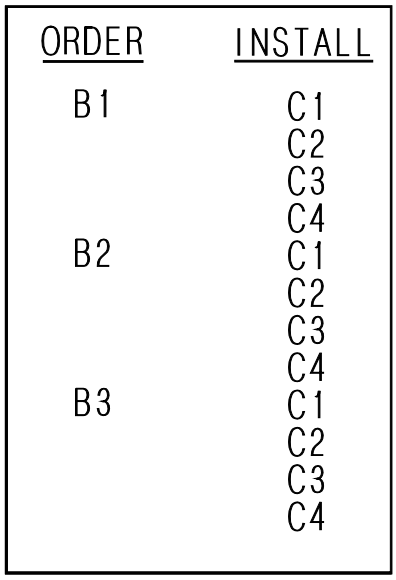



Figure 8-10 Report Output with Multi-Path Nested Looping

Although this is not necessarily the wanted results, the automatic looping process can be modified to produce the wanted results by specifying parallel looping for the report as described in the next section.

Multi-path nested looping is the default method of looping in a procedure, request, or report. No specification is necessary. It is activated automatically by VISION:Builder when you reference segment types from two or more different hierarchical legs.

When producing a subfile, multi-path nested looping is the default. (see Figure 8-11). Other entries are described later in this chapter. In fixed format syntax, multi-path nested looping occurs by leaving position [71] on the ER statement blank or by entering N.

```
FILE SUBF1 NAME CUSTSUB

SBNESTED: EXTRACT   FILE CUSTSUB,
          COLUMNS CUSTNO, CUSTNAME, ORDERNO, ORDRDATE, ORPERSON,
          INSTNO, INSTOPS
```

Figure 8-11 Multi-Path Nested Loop in the Subfile Output Command Group

**Note:** For a complete version of this example, see Multi-Path Nested Loop in the Subfile Output Command Group on page D-65.

## Multi-Path Parallel Looping

Based on the information in Multi-Path Nested Loops on page 8-8, it can be preferable on a report (or subfile) to suppress combinations of all possible segment occurrences in multi-path nested looping. The effect of this suppression is to output information in the segments just once, producing a report in abbreviated format. VISION:Builder accomplishes this by initiating a multi-path parallel loop, rather than a multi-path nested loop. This specification is entirely under your control for any given report or subfile. It is one of the means by which you can control the looping in VISION:Builder.

Figure 8-12 illustrates the flow of control in executing loops in parallel. In parallel looping, the first statement that initiates a loop, in effect, initiates the innermost loop. This is the reverse of multi-path nested looping. However, there is one important difference. When moving to the next segment occurrence in a loop, VISION:Builder moves to the next segment occurrence in all other loops as well

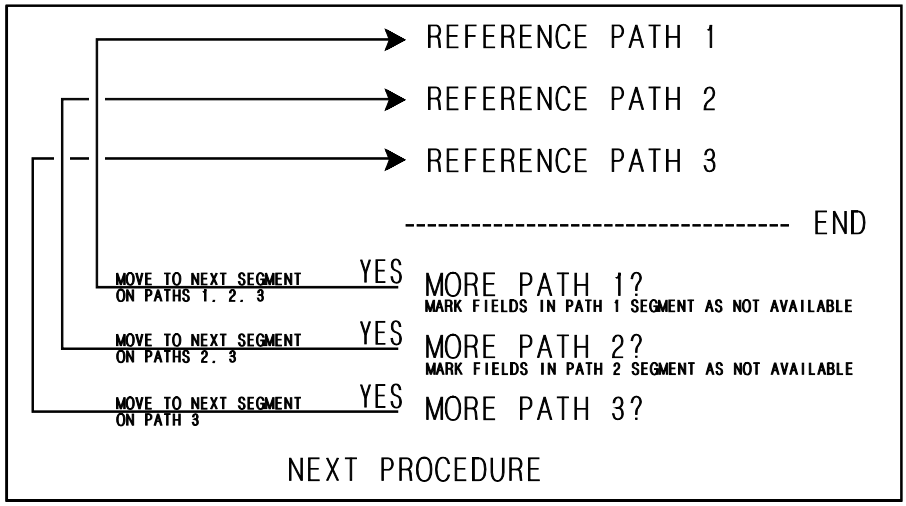



Figure 8-12 Multi-Path Parallel Loops Flow of Control

To produce a subfile with multi-path parallel looping, use the ABBREVIATED keyword on the EXTRACT command (see Figure 8-13). The following discussion on how multi-path parallel looping affects reports applies to subfiles as well. In fixed format syntax, specify multi-path parallel looping by entering R in position [71] of the ER statement.

```
FILE SUBF1 NAME CUSTSUB

SBPARALL: EXTRACT   FILE CUSTSUB, ABBREVIATED,
          COLUMNS CUSTNO, CUSTNAME, ORDERNO, ORDRDATE, ORPERSON,
          INSTNO, INSTOPS
```

Figure 8-13 Multi-Path Parallel Looping in the Subfile Output Command Group

**Note:** For a complete version of this example, see Multi-Path Nested Loop in the Subfile Output Command Group on page D-65.

You can process reports in multi-path parallel or multi-path nested looping. To specify multi-path parallel looping, use the ABBREVIATED keyword on the REPORT command in the Report command group (see Figure 8-14). The default is multi-path nested looping. Any loops initiated in the calling procedure remain as nested loops and are not affected by this specification.

```
RPPARALL: REPORT  ABBREVIATED,
          COLUMNS CUSTNO, CUSTNAME, ORDERNO, ORDRDATE, ORPERSON,
          INSTNO, INSTOPS
          TITLE 'SIMPLE REPORT USING MULTI-PATH PARALLEL LOOPING'
    END REPORT
```

Figure 8-14 Multi-Path Parallel Looping in a Report in the Report Command Group

**Note:** For a complete version of this example, see Multi-Path Nested Loop in the Subfile Output Command Group on page D-65.

Using the same file structure in Figure 8-9, contrast the effects of multi-path nested and parallel looping on the resulting report. The same effects would be seen for a subfile. Figure 8-10 shows the report produced with nested looping. If you make one change in the report and specify ABBREVIATED format, the report now appears as shown in Figure 8-15. The INSTALL segment information is no longer repeated. A dash prints when there are no more occurrences of the ORDER segment.

| ORDER | INSTALL |
|---|---|
| B1 | C1 |
| B2 | C2 |
| B3 | C3 |
| - | C4 |

Figure 8-15 Report Output with Multi-Path Parallel Looping (Abbreviated Format Report)

## Examples of Single-Path Nested Loops

The following examples use the structure of the customer file shown in Figure 8-1.

### Example 1

Figure 8-16 through Figure 8-19 illustrate a single-path nested loop started in a procedure statement.

Figure 8-16 shows the Master File Record Control (Main) procedure definition.

```
MAIN: PROC
;Each Master-File-Record Control Procedure
    CALL PROCLOOP
END PROC
```

Figure 8-16 Loop Procedure Invoked by Master File Record Control (Main)

**Note:** For a complete version of this example, see Multi-Path Nested Loop in the Subfile Output Command Group on page D-65.

Figure 8-17 shows the procedure definition.

The first IF statement excludes records for customer 00005.

The first reference to ORDRDATE in line A020 of the procedure starts a lower level loop on the ORDER segment. This statement tests for validity and existence of the ORDRDATE field.

The CALL statement is used to generate a report (REPORT1).

```
PROCLOOP: PROC TYPE SUBROUTINE
    IF CUSTNO NE '00005'
A020:  IF ORDRDATE EQ ORDRDATE
          CALL REPORT REPORT1  ;invokes report with looping
       END
    END
END
```

Figure 8-17 Loop Initiated in Procedure PROCLOOP

**Note:** For a complete version of this example, see Multi-Path Nested Loop in the Subfile Output Command Group on page D-65.

Figure 8-18 shows the report definition. Because a loop has already been initiated for the ORDER segment, no new loops are begun in this report.

```
REPORT1: REPORT  TYPE SUBROUTINE,
         COLUMNS CUSTNAME, CUSTNO, ORDERNO, ORDRDATE, ORPERSON
         TITLE 'LOOP DOMAIN LIMITED BY PROCEDURE PROCLOOP'
         END REPORT
```

Figure 8-18 Report Command Group – Loop Initiated in Procedure PROCLOOP

**Note:** For a complete version of this example, see Multi-Path Nested Loop in the Subfile Output Command Group on page D-65.

As seen in the report displayed in Figure 8-19, records without an ORDRDATE are not printed. This is because the test to see if ORDRDATE is valid and exists (line A020) fails and REPORT1 is not called. See Chapter 4, *Field Characteristics and Handling* for a discussion of invalid and missing fields.

```
                                    LOOP DOMAIN LIMITED BY PROCEDURE PROCLOOP                          PAGE 1
-------------------------------------------------------------------------------------------------------------------
          CUSTOMER            CUSTOMER  ORDER   ORDER   SALES
            NAME               NUMBER   NUMBER  DATE    PERSON
-------------------------------------------------------------------------------------------------------------------
 FINANCIAL SERVICES             00002    20103   033101   JJK
 FINANCIAL SERVICES             00002    20105   091501   MBK
 THE INSURANCE COMPANY          00003    30101   052301   JCR
 HOME IMPROVEMENT SUPPLIES      00010    01101   011301   GOA
 HOME IMPROVEMENT SUPPLIES      00010    01102   011301   TAH
 HOME IMPROVEMENT SUPPLIES      00010    01103   021501   TAH
 HOME IMPROVEMENT SUPPLIES      00010    01104   021501   JJK
```

Figure 8-19 Resulting Report when Loop Initiated in a Procedure

**Note:** For a complete version of this example, see Multi-Path Nested Loop in the Subfile Output Command Group on page D-65.

### Example 2

Figure 8-20 through Figure 8-22 show the same procedure PROCLOOP calling another report (REPORT1A). REPORT1A does not reference any lower level segments. This example illustrates that the range of a loop started in a procedure statement extends to the report that it calls.

```
PROCLOOP: PROC TYPE SUBROUTINE
    IF CUSTNO NE '00005'
A020:  IF ORDRDATE EQ ORDRDATE
          CALL REPORT REPORT1  ;invokes report with looping
          CALL REPORT REPORT1A ;invokes report with limited range
       END
    END
END
```

Figure 8-20 Loop initiated in Procedure PROCLOOP

**Note:** For a complete version of this example, see Multi-Path Nested Loop in the Subfile Output Command Group on page D-65.

```
REPORT1A: REPORT  TYPE SUBROUTINE,
          COLUMNS CUSTNAME, CUSTNO
          TITLE 'LOOPS INITIATED IN PROCEDURE'
          TITLE 'RANGE LIMITED IN REPORT'
    END REPORT
```

Figure 8-21 Report Command group – Loop Initiated in Procedure PROCLOOP – Limited in Report

**Note:** For a complete version of this example, see Multi-Path Nested Loop in the Subfile Output Command Group on page D-65.

```
                                        LOOPS INITIATED IN PROCEDURE                                    PAGE 1
                                          RANGE LIMITED IN REPORT
-------------------------------------------------------------------------------------------------------------------------
          CUSTOMER            CUSTOMER
            NAME               NUMBER
-------------------------------------------------------------------------------------------------------------------------
 FINANCIAL SERVICES             00002
 FINANCIAL SERVICES             00002
 THE INSURANCE COMPANY          00003
 HOME IMPROVEMENT SUPPLIES      00010
 HOME IMPROVEMENT SUPPLIES      00010
 HOME IMPROVEMENT SUPPLIES      00010
 HOME IMPROVEMENT SUPPLIES      00010
```

Figure 8-22 Resulting Report when Loop Initiated in a Procedure and Limited in Report

**Note:** For a complete version of this example, see Multi-Path Nested Loop in the Subfile Output Command Group on page D-65.

Report REPORT1A references no fields in lower level segments (see Figure 8-21). However, because a loop was already in effect due to statement A020, the report (Figure 8-22) prints one line for each ORDER segment occurrence. This is an important concept because in a given request, a loop started in a procedure statement extends to the report it calls.

## Example 3

Figure 8-23 and Figure 8-24 show a different procedure (PROCLUP3) with a loop begun in a report.

```
PROCLUP3: PROC TYPE SUBROUTINE
    IF CUSTNO NE '00005' AND CUSTNO NE '00015'
          CALL REPORT REPORT2  ;  Only the root segment is referenced
          CALL REPORT REPORT4  ;  in the procedure.  All loops are
          CALL REPORT REPORT5  ;  initiated in the reports.
    END
END

REPORT2: REPORT  TYPE SUBROUTINE,
         COLUMNS CUSTNAME, CUSTNO, ORDERNO, ORDRDATE, ORPERSON
         TITLE 'LOOP STARTS IN REPORT'
         END REPORT
```

Figure 8-23 Loops initiated in Report Command Group

**Note:** For a complete version of this example, see Multi-Path Nested Loop in the Subfile Output Command Group on page D-65.

```
                                        LOOP STARTS IN REPORT                                           PAGE 1
-------------------------------------------------------------------------------------------------------------------------
          CUSTOMER            CUSTOMER  ORDER   ORDER   SALES
            NAME               NUMBER   NUMBER  DATE    PERSON
-------------------------------------------------------------------------------------------------------------------------
 FINANCIAL SERVICES             00002    20103   033101   JJK
 FINANCIAL SERVICES             00002    20105   091501   MBK
 THE INSURANCE COMPANY          00003    30101   052301   JCR
 HOME IMPROVEMENT SUPPLIES      00010    01101   011301   GOA
 HOME IMPROVEMENT SUPPLIES      00010    01102   011301   TAH
 HOME IMPROVEMENT SUPPLIES      00010    01103   021501   TAH
 HOME IMPROVEMENT SUPPLIES      00010    01104   021501   JJK
```

Figure 8-24 Resulting Report when Loop Initiated in the Report Command Group

**Note:** For a complete version of this example, see Multi-Path Nested Loop in the Subfile Output Command Group on page D-65.

### Example 4

Figure 8-25 and Figure 8-26 show the effect of nested loops down a single-path of the hierarchy. The fields are referenced in hierarchical order: level 2 (ORDERNO) followed by level 3 (SHIPNO). REPORT4 is called by the procedure (PROCLUP3) shown in the last example (see Figure 8-23.

```
REPORT4: REPORT  TYPE SUBROUTINE,
         COLUMNS CUSTNAME, CUSTNO, ORDERNO, ORDRDATE, ORPERSON,
               SHIPNO, SHIPDATE
         TITLE 'SINGLE-PATH NESTED LOOP STARTED IN REPORT'
         TITLE 'HIERARCHIACAL ORDER'
         END REPORT
```

Figure 8-25 Single-Path Nested Loop -Hierarchical Order

**Note:** For a complete version of this example, see Multi-Path Nested Loop in the Subfile Output Command Group on page D-65.

SINGLE-PATH NESTED LOOP STARTED IN REPORT PAGE 1
HIERARCHIACAL ORDER

| CUSTOMER NAME | CUSTOMER NUMBER | ORDER NUMBER | ORDER DATE | SALES PERSON | SHIPPING NUMBER | DATE SHIPPED |
|---|---|---|---|---|---|---|
| FINANCIAL SERVICES | 00002 | 20103 | 033101 | JJK | D25C | 042501 |
| FINANCIAL SERVICES | 00002 | 20105 | 091501 | MBK | I15J | 091501 |
| THE INSURANCE COMPANY | 00003 | 30101 | 052301 | JCR | E28X | 052801 |
| HOME IMPROVEMENT SUPPLIES | 00010 | 01101 | 011301 | GOA | A25G | 012501 |
| HOME IMPROVEMENT SUPPLIES | 00010 | 01102 | 011301 | TAH | - | - |
| HOME IMPROVEMENT SUPPLIES | 00010 | 01103 | 021501 | TAH | - | - |
| HOME IMPROVEMENT SUPPLIES | 00010 | 01104 | 021501 | JJK | B19K | 021901 |

Figure 8-26 Resulting Report Single-Path Nested Loop -Hierarchical Order

**Note:** For a complete version of this example, see Multi-Path Nested Loop in the Subfile Output Command Group on page D-65.

REPORT4 starts an outer loop on the ORDER segment at the reference to the ORDERNO field. The second or inner loop starts on the SHIPINV segment at the reference to the SHIPNO field because it is referenced after ORDERNO. Note that the report shows no shipping numbers for order number 01102 and 01103.

### Example 5

Figure 8-27 and Figure 8-28 show the effect of referencing fields in an inverse hierarchical order for the same procedure (PROCLUP3) using a new report (REPORT5). The SHIPNO field (level 3) is referenced first, which starts a loop on the SHIPINV segment. The ORDRDATE field (level 2) is referenced next.

```
REPORT5: REPORT  TYPE SUBROUTINE,
         COLUMNS SHIPNO, SHIPDATE,
               ORDERNO, ORDRDATE, ORPERSON, CUSTNAME, CUSTNO
         TITLE 'SINGLE-PATH NESTED LOOP STARTED IN REPORT'
         TITLE 'INVERSE HIERARCHIACAL ORDER'
         END REPORT
```

Figure 8-27 Single-Path Nested Loop –Inverse Hierarchical Order

**Note:** For a complete version of this example, see Multi-Path Nested Loop in the Subfile Output Command Group on page D-65.

```
                                    SINGLE-PATH NESTED LOOP STARTED IN REPORT                    PAGE 1
                                         INVERSE HIERARCHIACAL ORDER

 SHIPPING   DATE     ORDER   ORDER   SALES          CUSTOMER          CUSTOMER
  NUMBER  SHIPPED  NUMBER   DATE   PERSON            NAME             NUMBER
-------------------------------------------------------------------------------------------------------

   D25C   042501   20103   033101   JJK   FINANCIAL SERVICES          00002
   I15J   091501   20105   091501   MBK   FINANCIAL SERVICES          00002
   E28X   052801   30101   052301   JCR   THE INSURANCE COMPANY       00003
   A25G   012501   01101   011301   GOA   HOME IMPROVEMENT SUPPLIES   00010
   B19K   021901   01104   021501   JJK   HOME IMPROVEMENT SUPPLIES   00010
```

Figure 8-28 Resulting Report Single-Path Nested Loop – Inverse Hierarchical Order

**Note:** For a complete version of this example, see Multi-Path Nested Loop in the Subfile Output Command Group on page D-65.

REPORT5 is in inverse hierarchical order. The first field referenced is SHIPNO, which starts a loop on the SHIPINV segment.

The second segment field referenced is ORDERNO, which is a higher level segment or a parent segment. Here, loops are generated by going up the leg of the structure.

Notice the effect of this in the report generated. Check the differences between this and the previous report. For example, order numbers 01102 and 01103 are is not reported. This is because the ITEMORD segment is the outermost loop and only orders that have at least one item are reported.

## Examples of Multi-Path Looping

The following examples contrast multi-path nested looping with multi-path parallel looping. Example 1 illustrates multi-path nested looping. Example 2 illustrates the effects of specifying Abbreviated Format (multi-path parallel looping).

### Example 1

This example shows multi-path nested looping in a report. Figure 8-29 shows the procedure (PRLOOP67) that calls the reports. Figure 8-30 shows the Report command group for REPORT6.

```
PRLOOP67: PROC TYPE SUBROUTINE
    IF CUSTNO NE '00005' AND CUSTNO NE '00015'
          CALL REPORT REPORT6  ;  report uses multi path nested loops
          CALL REPORT REPORT7  ;  report uses multi path parallel loops
    END
END
```

Figure 8-29 Multi-Path Loops: Procedure Statements

**Note:** For a complete version of this example, see Multi-Path Nested Loop in the Subfile Output Command Group on page D-65.

```
REPORT6: REPORT  TYPE SUBROUTINE,
         COLUMNS CUSTNAME, CUSTNO, ORDERNO, ORDRDATE, ORPERSON,
               INSTNO, INSTOPS
         ORDER BY CUSTNAME, CUSTNO
         GROUP BY CUSTNAME, CUSTNO
         TITLE 'MULTI-PATH NESTED LOOPING'
         END REPORT
```

Figure 8-30 Multi-Path Nested Loop: Report Command Group

**Note:** For a complete version of this example, see Multi-Path Nested Loop in the Subfile Output Command Group on page D-65.

In the Report command group (Figure 8-30), multi-path nested loops are started on the ORDER and INSTALL segments. The reference to the ORDERNO field starts an outer loop on the ORDER segment. The reference to the INSTNO field starts an inner loop on the INSTALL segment. This produces the report in Figure 8-31. Notice that the installation numbers are repeated for each new order number.

```
                                        MULTI-PATH NESTED LOOPING                                    PAGE 1
-------------------------------------------------------------------------------------------------------------------------
          CUSTOMER            CUSTOMER  ORDER   SALES   INSTALLATION  OPERATING
            NAME               NUMBER   NUMBER  PERSON     NUMBER      SYSTEM
-------------------------------------------------------------------------------------------------------------------------
 FINANCIAL SERVICES             00002    20103   JJK        A031       OS/390
                                         20105   MBK        A031       OS/390
 HOME IMPROVEMENT SUPPLIES      00010    01101   GOA        D432       OS/390
                                         01102   TAH        D432       OS/390
                                         01103   TAH        D432       OS/390
                                         01104   JJK        D432       OS/390
 THE INSURANCE COMPANY          00003    30101   JCR        B051       VSE/ESA
                                         30101   JCR        G077       OS/390
                                         30101   JCR        M452       VM/CMS
                                         30101   JCR        R802       OS/390
                                         30102   TAH        B051       VSE/ESA
                                         30102   TAH        G077       OS/390
                                         30102   TAH        M452       VM/CMS
                                         30102   TAH        R802       OS/390
```

Figure 8-31 Resulting Report Multi-Path Nested Loop

**Note:** For a complete version of this example, see Multi-Path Nested Loop in the Subfile Output Command Group on page D-65.

### Example 2

This example shows multi-path parallel looping in a report. The same example shown in Figure 8-29 and Figure 8-30 is used with one alteration. As shown in Figure 8-32, the ABBREVIATED keyword has been added to the REPORT statement.

```
REPORT7: REPORT  TYPE SUBROUTINE, ABBREVIATED,
         COLUMNS CUSTNAME, CUSTNO, ORDERNO, ORPERSON,
               INSTNO, INSTOPS
         ORDER BY CUSTNAME, CUSTNO
         GROUP BY CUSTNAME, CUSTNO
         TITLE 'MULTI-PATH PARALLEL LOOPING'
         END REPORT
```

Figure 8-32 Multi-Path Parallel Loop: Report Command Group

**Note:** For a complete version of this example, see Multi-Path Nested Loop in the Subfile Output Command Group on page D-65.

With this specification alone, multi-path parallel loops (rather than nested loops) are started on the ORDER and INSTALL segments in REPORT7 (see Figure 8-33). The reference to the ORDERNO field starts a loop on the ORDER segment. The reference to the INSTNO field starts a parallel loop on the INSTALL segment. This produces parallel loops across the paths when the lower level segment types are referenced. Any loops that may have been initiated in the calling procedure remain as nested loops and are not affected by this specification.

This produces the report shown in Figure 8-33. Notice that installations are reported just once. They are not repeated for each new order number. This is especially apparent for customer 00003. The dashes for customer 00003 occur because VISION:Builder continues looping on installations when it runs out of orders. VISION:Builder prints missing occurrences with dashes or other installation-specified character.

MULTI-PATH PARALLEL LOOPING PAGE 1

| CUSTOMER NAME | CUSTOMER NUMBER | ORDER NUMBER | SALES PERSON | INSTALLATION NUMBER | OPERATING SYSTEM |
|---|---|---|---|---|---|
| FINANCIAL SERVICES | 00002 | 20103 | JJK | A031 | OS/390 |
| | | 20105 | MBK | - | - |
| HOME IMPROVEMENT SUPPLIES | 00010 | 01101 | GOA | D432 | OS/390 |
| | | 01102 | TAH | - | - |
| | | 01103 | TAH | - | - |
| | | 01104 | JJK | - | - |
| THE INSURANCE COMPANY | 00003 | 30101 | JCR | B051 | VSE/ESA |
| | | 30102 | TAH | G077 | OS/390 |
| | | - | - | M452 | VM/CMS |
| | | - | - | R802 | OS/390 |

Figure 8-33 Resulting Report Multi-Path Parallel Loop

**Note:** For a complete version of this example, see Multi-Path Nested Loop in the Subfile Output Command Group on page D-65.

## Example 3

This example illustrates that any loops initiated in a procedure remain nested and are not affected by the ABBREVIATED keyword (parallel looping). Example 2, which used abbreviated format, is the base for this example. However, this example uses the PRL00P8 procedure instead of PRLOOP67 to call the report. As shown in Figure 8-34, PRLOOP8 initiates a nested loop on the ORDER segment by referencing ORDERNO in the IF statement. Then, it calls the report REPORT8. However, no new loop is initiated on the ORDER segment in REPORT8 because the loop was already in effect in the procedure. REPORT8 initiates a new loop on

the INSTALL segment only, resulting in multi-path nested looping. The resulting report is the same as shown in Figure 8-31 on page 8-18 with the installation repeated for each order number.

```
PRLOOP8: PROC TYPE SUBROUTINE
    IF CUSTNO NE '00005' AND CUSTNO NE '00015'
      IF ORDERNO EQ ORDERNO        ; nested loop initiated here
         CALL REPORT REPORT8       ; will override 'ABBREVIATED'
      END                          ; keyword in REPORT8
    END
END

REPORT8: REPORT  TYPE SUBROUTINE, ABBREVIATED,
         COLUMNS CUSTNAME, CUSTNO, ORDERNO, ORPERSON,
               INSTNO, INSTOPS
         ORDER BY CUSTNAME, CUSTNO
         GROUP BY CUSTNAME, CUSTNO
         TITLE 'MULTI-PATH PARALLEL LOOPING'
         END REPORT
```

Figure 8-34 Multi-Path Parallel Loop Specified in Report – Nested Loop Initiated in Procedure

**Note:** For a complete version of this example, see Multi-Path Nested Loop in the Subfile Output Command Group on page D-65.

# Selection Control Looping

Selection control is another means by which you can control the looping in VISION:Builder. Sometimes it is necessary to report information from higher level segments based upon conditions in lower level segments. However, each record is to be reported only once (the first selection of the record). This can be achieved by coding SELECTION_CONTROL YES on the REPORT command. With selection control, processing of the current record terminates once the selection criteria in the processing logic are satisfied for one or more segment types.

Figure 8-35 shows how this command provides the selection criterion for each record.

```
SELECT: REPORT  SELECTION_CONTROL YES,
         COLUMNS CUSTNAME, CUSTNO, ORDERNO, ORPERSON,
               INSTNO, INSTOPS
         ORDER BY CUSTNAME, CUSTNO
         GROUP BY CUSTNAME, CUSTNO
         TITLE 'LOOPING LIMITED BY SELECTION CONTROL'
         END REPORT
```

Figure 8-35 Loop Limited by Selection Control

**Note:** For a complete version of this example, see Multi-Path Nested Loop in the Subfile Output Command Group on page D-65.

Figure 8-36 shows the resulting report.

LOOPING LIMITED BY SELECTION CONTROL PAGE 1

| CUSTOMER NAME | CUSTOMER NUMBER | ORDER NUMBER | SALES PERSON | INSTALLATION NUMBER | OPERATING SYSTEM |
|---|---|---|---|---|---|
| CLOTHES LINE INC | 00015 | - | - | E919 | OS/390 |
| FINANCIAL SERVICES | 00002 | 20103 | JJK | A031 | OS/390 |
| HOME IMPROVEMENT SUPPLIES | 00010 | 01101 | GOA | D432 | OS/390 |
| RECREATIONAL TOYS | 00005 | - | - | C513 | VM/CMS |
| THE INSURANCE COMPANY | 00003 | 30101 | JCR | B051 | VSE/ESA |

Figure 8-36 Resulting Report Selection Control

**Note:** For a complete version of this example, see Multi-Path Nested Loop in the Subfile Output Command Group on page D-65.

Only the first order for each customer is reported. This is because selection control means no looping after selection in the processing logic is satisfied. Because there was no filter in previous procedure logic, the first occurrence of each referenced segment type is available for output.

# Segment Operators in Looping

There are two ASL statements used in controlling automatic looping.

- FIND (find segment function) allows you to get a specific occurrence of a lower level segment without looping through every one.
- RELEASE (release segment command) returns a segment referenced by the FIND statement to an unreferenced status. A loop can be started on that segment or its dependents as early as the next statement.

The FIND function and the RELEASE command are used in procedure statements and save considerable processing overhead by restricting access to specific segments.

For more information about ASL, refer to the *VISION:Builder ASL Reference Guide*.

## FIND

The FIND function allows direct control over the segment it references. The FIND function is a conditional function used in a logical expression to find an occurrence of a segment. The FIND function can cause the first, last, or next occurrence of a lower level segment to be located for processing. The FIND function can specify a constant or any field. If a field is specified, the search takes place until an occurrence is found that is equal to the current value in the field. If the field is in a lower level segment, an automatic loop is established on that lower level segment and all the VISION:Builder rules for automatic looping apply.

For example, use FIND to get the lower level segment ORDER of the CUSTOMER file:

```
IF FIND(ORDER WHERE ORDERNUM = '01104')
   CALL REPORT9
END
```

This results in the report shown in Figure 8-37.

```
                                                FIND ORDER 01104                                                PAGE 1
-------------------------------------------------------------------------------------------------------------------------
            CUSTOMER            CUSTOMER  ORDER   SALES   ORDER
              NAME               NUMBER   NUMBER  PERSON  DATE
-------------------------------------------------------------------------------------------------------------------------
  HOME IMPROVEMENT SUPPLIES       00010    01104    JJK    021501
```

Figure 8-37 Resulting Report - Qualified FIND

**Note:** For a complete version of this example, see Multi-Path Nested Loop in the Subfile Output Command Group on page D-65.

VISION:Builder searches for the first occurrence of the ORDER segment with a key value equal to 01104. When found, only this occurrence is available for further processing in this procedure.

If you use the FIND function without a selection expression, it is an unqualified FIND; it will get the next occurrence of the referenced segment. For example:

```
IF FIND(SEGMENT ORDER)
  CALL REPORT REP10
END
```

Figure 8-38 shows that the first occurrence of the ORDER segment is selected for each customer where ORDER segments exist. The specification of a second unqualified FIND on the ORDER segment would access the next occurrence of the ORDER segment.

```
                                                UNQUALIFIED FIND                                                PAGE 1
-------------------------------------------------------------------------------------------------------------------------
            CUSTOMER            CUSTOMER  ORDER   SALES   ORDER
              NAME               NUMBER   NUMBER  PERSON  DATE
-------------------------------------------------------------------------------------------------------------------------
  FINANCIAL SERVICES              00002    20103    JJK    033101
  THE INSURANCE COMPANY           00003    30101    JCR    052301
  HOME IMPROVEMENT SUPPLIES       00010    01101    GOA    011301
```

Figure 8-38 Resulting Report - Unqualified FIND

**Note:** For a complete version of this example, see Multi-Path Nested Loop in the Subfile Output Command Group on page D-65.

The following lists the rules associated with the FIND function as it applies to segment manipulation:

- When the FIND function is applied to a lower level segment, the parent is placed in an automatic loop, if it is not already in a loop.
- The FIND function on a segment with multiple keys uses only the major key. If the major key is not unique, only the first matching segment is available.
- When the FIND function cannot find the specified occurrence, it is said to fail. In this case, fields in the referenced segment type and all its dependent segments are considered missing (indicated by dashes in reports and considered unequal in compares and invalid in arithmetic results).
- The FIND function is suppressed if it references segment types already part of a loop or whose dependents are part of a loop. This includes a FIND function (or RELEASE command) in subroutine procedures and requests that attempt to alter loops they inherit from the calling procedure. An operation that is suppressed is, in effect, ignored by the procedure and processing continues with the next statement. The current segment occurrence is available for further processing.
- A qualified FIND function that specifies a field in a previously unreferenced lower level segment as its key value can establish a loop on the occurrences of that segment type.
- A FIND function specified for a segment of an IMS file that has preselection operates only on those segments first satisfied by the preselection criteria.
- A RELEASE command on segment types (or their dependents) in an automatic loop is suppressed.
- A successful RELEASE command places the referenced segment type in an unreferenced status. The segment type is available for automatic looping as though it was never referenced by a FIND function.

For more information about the FIND function, see the *VISION:Builder ASL Reference Guide*.

## RELEASE

The RELEASE command returns a segment referenced by a FIND function to an unreferenced status. A loop can be started on that segment type or its dependents as early as the next statement. After the RELEASE command in the procedure, any reference to the segment or array will result in positioning at the beginning of the segments or the beginning of the array. You cannot specify the name of a root segment in the RELEASE command.

Example:

```
IF FIND(SEGMENT ORDER WHERE ORDERNO=10001)
  RELEASE SEGMENT ORDER
  CALL REPORT ORDERS
END
```

The FIND function would be true if there is at least one segment with an order number of 10001. The function positions on the first segment that is found. The RELEASE command releases the ORDER segment found and the report lists all of the current orders. Without the RELEASE command, the report ORDERS would only access the single segment established by the FIND function.

Note that it is possible to go back to previously processed segments by using the FIND function and the RELEASE command.

For the FIND function and the RELEASE command, VISION:Builder supplies an internal flag field, the LSTATUS flag.

For more information about the RELEASE command, refer to the *VISION:Builder ASL Reference Guide*.

### The Logical Status Flag (LSTATUS Flag)

The LSTATUS flag is a four-byte character string that contains a status value following the execution of any segment function or command. This flag and its contents are independent of the type of file (for example, sequential, and IMS).

As part of any segment operation, there is an implied compare to blanks in the LSTATUS flag. Thus, an IF FIND statement will be false for any status value other than blanks. You can examine status information to determine why the operation failed or was suppressed. Refer to the *VISION:Builder ASL Reference Guide* for more information about the values of the LSTATUS flag.

## Subroutine Looping

This section discusses how loops are initiated when procedures and subroutine procedures are involved. Each subroutine procedure or subroutine request inherits the loop from the calling procedure or request. On completion of processing, control is returned to the calling procedure or request.

Subroutine procedures cannot change a loop they inherit. This means they cannot perform the FIND function and the RELEASE command on segment types they inherit from the calling procedure or request. However, subroutines can perform the following functions:

- Start a new loop by referencing a field in a segment type not inherited from the calling procedure or request
- Perform FIND functions and RELEASE commands on segment types not inherited from the calling procedure or request.

These functions and commands are performed for the duration of that subroutine only and do not pass any new loops back to the calling procedure or request.

If the FIND function and the RELEASE command are attempted in subroutines on loops (segment types) inherited from the calling procedure or request, the LSTATUS flag is set to INHR and the operation is suppressed (ignored).

Figure 8-39 illustrates some cases of correct and incorrect use of segment operations used in subroutine procedures.

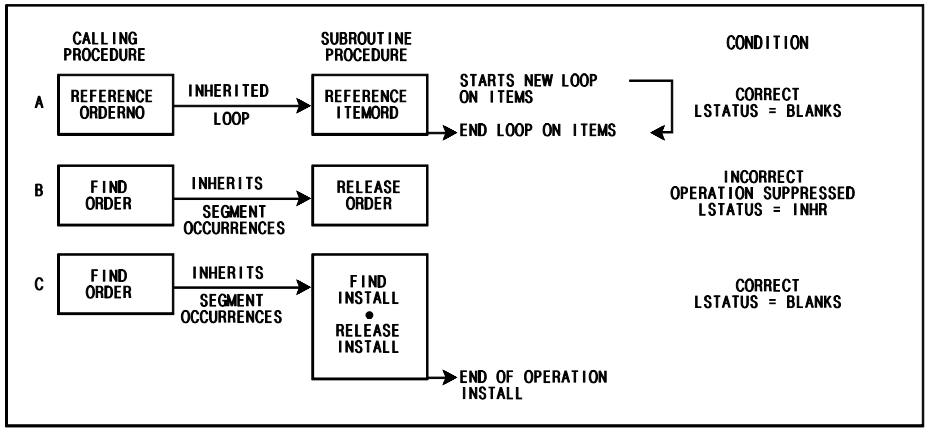



Figure 8-39 Subroutine Looping

In case A, an automatic loop is started on the ORDER segment that is inherited by the subroutine. The subroutine starts a new loop (inner loop) on the dependent segment ITEMORD. The loop on ITEMORD is executed for each segment occurrence inherited from the calling procedure. The loop on ITEMORD ends in the subroutine and is not passed back to the calling procedure. However, the outer loop on ORDER is passed back to the calling procedure.

In case B, a FIND function on the ORDER segment in the calling procedure selects the first occurrence on ORDER and passes it to the subroutine procedure. The subroutine procedure attempts to do a segment command (RELEASE) on the same segment type it inherited. This causes the LSTATUS flag to be set to INHR and the operation is suppressed (ignored). The segment occurrence is returned to the calling procedure.

In case C, the FIND function on the ORDER segment produces the first segment occurrence on ORDER that is inherited by the subroutine procedure. The subroutine procedure issues two segment functions on the INSTALL segment. Because the INSTALL segment is neither inherited nor in an automatic loop, the functions are successful.

The segment functions on the INSTALL segment extend only as far as the end of the subroutine procedure. The ORDER segment occurrence is passed back to the calling procedure.

It is important to note that a field that becomes invalid in a calling procedure can be passed on to a subroutine procedure as an invalid field. Similarly, invalid fields can be propagated from the subroutine to the calling procedure.

# Branching to Control Looping

Automatic looping can also be controlled with the use of VISION:Builder branching capabilities. The following table lists the effects on looping for the corresponding ASL and fixed format syntax branch locations

| ASL Syntax | Fixed Format Syntax | Effect on Looping |
|---|---|---|
| N/A | GO NEXT REQUEST | Terminates all processing of the current request for the current record and transfers control to the next request. All looping on all segment occurrences in the current request stops for that record and the next request is processed. |
| CONTINUE | GO END | Terminates looping on the current segment occurrence and gets the next one. |
| TRANSFER NEXT MASTER | GO NEXT MASTER | Terminates all processing of the current record for all procedures and requests, including subroutine procedures. Control is transferred to the next VISION:Builder function. All looping for the current record stops for all procedures and requests. |
| N/A | GO REQUEST B | Terminates all processing of the current request for the current record and transfers control to request B. All looping on all segment occurrences in the current request stops for that record and Request B is processed. |
| N/A | GO OUTPUT | Terminates looping on the current segment occurrence, outputs it, and gets the next occurrence, if any. |
| RETURN or LEAVE | GO RETURN | Terminates all processing of the current subroutine procedure, subroutine request, or subroutine set. It returns to the statement following the one that called the subroutine procedure, subroutine request, or subroutine set. |

In addition, in fixed format syntax, a backward branch to a procedure or request before the current one can be taken. In this case, the effect on looping is the same as that of a forward branch.

For details on fixed format syntax for requests, subroutines, subroutine sets, and request branching, see Chapter 11, *Requests*. For details on branch locations in ASL procedures, refer to the *VISION:Builder ASL Reference Guide*.

Chapter

# 9 Reporting

This chapter describes various reporting concepts by using examples of reporting from an inventory file.

## Definition of the Inventory File

The inventory file is a structured file. It contains distinct levels of information: the item level and the backorder level. Each item can have many backorders. This structure allows related and repetitive data to be combined into one logical record, simplifying the processing and maintaining of data.

The definitions for the inventory file definition, ITEMFILE, are shown in Figure 9-1 and Figure 9-2.

```
                                    FILE STRUCTURE DIAGRAM FOR                    DIAGRAM PAGE  1
                                    FILE DEFINITION - ITEMFILE

RECORD FORMAT = KEY SEQUENCED VSAM                                  FILE IDENTIFICATION =
RECORD SIZE = 80      BUFFER SIZE = 80                              NUMBER OF SEGMENTS IN FILE = 2

                                         +---------+
                                         °ITEMSEG  °
                                         +---------+
                                              °
                                              °
                                              °
                                         +---------+
                                         °BKORDSEG °
                                         +---------+
```

Figure 9-1 File Structure for ITEMFILE Inventory File

```
                                  DETAILED GLOSSARY BY NAME FOR                   GLOSSARY PAGE 1
                                    FILE DEFINITION - ITEMFILE

DATE CATALOGED - 11/16/01 14:13:36                          DATE LAST UPDATED - 11/16/01 14:13:36
CATALOGED BY -                                              LAST UPDATED BY -
EXPIRATION DATE -                                           DATE LAST USED -
FILE IDENTIFICATION =                                       RECORD FORMAT = KEY SEQUENCED VSAM
NUMBER OF SEGMENTS IN FILE = 2                              RECORD SIZE = 80
NUMBER OF FIELDS IN FILE = 10                               BUFFER SIZE = 80


                                  ********************************
                                  *    SEGMENT   1, LEVEL 1     *
                                  *   SEGMENT NAME = ITEMSEG    *
                                  ********************************
SEGMENT OCCURS N TIMES = 1                        KEY FIELD 1 = ITEMNO     TYPE = C   LENGTH = 7
SEGMENT SIZE = 63
NUMBER OF FIELDS IN SEGMENT = 8
SEGMENT KEY CHARACTERISTICS = U
```

Figure 9-2 Glossary for ITEMFILE Inventory File (Page 1 of 2)

| FIELD NAME | FIELD TYPE | FIELD LOCATION | FIELD LENGTH | FIELD RNDING | DEC PLACES | CNT FIELD FOR SGMT | EDIT CODES ( ) ( ) ( ) | EDIT LENGTH | OUTPUT WIDTH | LINE NO | DATA BASE NAME/ *** COLUMN HEADING *** |
|---|---|---|---|---|---|---|---|---|---|---|---|
| ITEMNAME | C | 8 | 20 | | | | | 20 | 20 | | |
| | | | | | | | | | | 1 | *** ITEM *** |
| | | | | | | | | | | 2 | *** NAME *** |
| ITEMNO | C | 1 | 7 | | | | | 7 | 7 | | |
| | | | | | | | | | | 1 | *** ITEM *** |
| | | | | | | | | | | 2 | *** NUMBER *** |
| ITMPRICE | Z | 28 | 6 | | 2 | | ($) ( ) ( ) | 9 | 9 | | |
| | | | | | | | | | | 1 | *** ITEM *** |
| | | | | | | | | | | 2 | *** PRICE *** |
| ORDQTY | Z | 42 | 4 | | | | | 6 | 6 | | |
| | | | | | | | | | | 1 | *** ORDER *** |
| | | | | | | | | | | 2 | *** QTY *** |
| QANONHND | Z | 39 | 3 | | | | | 4 | 6 | | |
| | | | | | | | | | | 1 | *** QTY ON *** |
| | | | | | | | | | | 2 | *** HAND *** |
| REORDPT | Z | 46 | 3 | | | | | 4 | 7 | | |
| | | | | | | | | | | 1 | *** REORDER *** |
| | | | | | | | | | | 2 | *** POINT *** |
| UNITCOST | Z | 34 | 5 | | 2 | | ($) ( ) ( ) | 7 | 7 | | |
| | | | | | | | | | | 1 | *** UNIT *** |
| | | | | | | | | | | 2 | *** COST *** |
| VENDOR | C | 49 | 15 | | | | | 15 | 15 | | |
| | | | | | | | | | | 1 | *** VENDOR *** |

DETAILED GLOSSARY BY NAME FOR FILE DEFINITION - ITEMFILE

GLOSSARY PAGE 2

**********
* SEGMENT 2, LEVEL 2 *
* SEGMENT NAME = BKORDSEG *
**********

SEGMENT OCCURS N TIMES = 2
SEGMENT SIZE = 8
NUMBER OF FIELDS IN SEGMENT = 2
SEGMENT KEY CHARACTERISTICS = U

KEY FIELD 1 = ORDERNO TYPE = C LENGTH = 5

| FIELD NAME | FIELD TYPE | FIELD LOCATION | FIELD LENGTH | FIELD RNDING | DEC PLACES | CNT FIELD FOR SGMT | EDIT CODES ( ) ( ) ( ) | EDIT LENGTH | OUTPUT WIDTH | LINE NO | DATA BASE NAME/ *** COLUMN HEADING *** |
|---|---|---|---|---|---|---|---|---|---|---|---|
| ORDERNO | C | 1 | 5 | | | | | 5 | 6 | | |
| | | | | | | | | | | 1 | *** ORDER *** |
| | | | | | | | | | | 2 | *** NUMBER *** |
| QTYBKORD | Z | 6 | 3 | | | | | 4 | 7 | | |
| | | | | | | | | | | 1 | *** QTY BACK *** |
| | | | | | | | | | | 2 | *** ORDERED *** |

Figure 9-2 Glossary for ITEMFILE Inventory File (Page 2 of 2)

# Generating Simple Reports

Figure 9-3 shows a sample inventory check list report. VISION:Builder can produce this report largely through automatic functions.

06/14/02 INVENTORY CHECKLIST PAGE 1

| ITEM NUMBER | ITEM NAME | QTY ON HAND | REORDER POINT | ORDER QTY | UNIT COST | ITEM PRICE | VENDOR | ORDER NUMBER | QTY BACK ORDERED |
|---|---|---|---|---|---|---|---|---|---|
| BL-B01 | BLDR QUICK REF | 249 | 100 | 200 | $7.00 | $7.95 | ACCESS MASTERS | - | - |
| BL-E68 | BUILDER CONCEPTS | 87 | 100 | 200 | 20.25 | 27.50 | ACCESS MASTERS | 45097 | 90 |
| BL-F43 | BLDR MESSAGES | 16 | 100 | 200 | 12.25 | 21.00 | ACCESS MASTERS | 10081 | 50 |
| BL-F43 | BLDR MESSAGES | 16 | 100 | 200 | 12.25 | 21.00 | ACCESS MASTERS | 31565 | 50 |
| BL-H11A | BLDR HANDBOOK ASL | 56 | 100 | 200 | 23.25 | 31.00 | ACCESS MASTERS | 10987 | 25 |
| BL-H11A | BLDR HANDBOOK ASL | 56 | 100 | 200 | 23.25 | 31.00 | ACCESS MASTERS | 12568 | 50 |
| BL-H11F | BLDR HANDBOOK FIXED | 50 | 50 | 100 | 23.75 | 35.00 | ACCESS MASTERS | 65012 | 60 |
| BL-T01B | BLDR TRAINING BASIC | 15 | 5 | 5 | .00 | 775.00 | IN HOUSE | 90311 | 2 |
| BL-T02A | BLDR TRAINING ADV | 15 | 5 | 5 | .00 | 1,000.00 | IN HOUSE | - | - |
| GN-N34 | DATA BASE CONCEPTS | 14 | 50 | 50 | 33.25 | 39.95 | TCOM CONSULTING | - | - |
| IN-E13 | INFORM USER GUIDE | 175 | 100 | 200 | 23.25 | 28.95 | QUERY PRO | 11011 | 15 |
| IN-G61 | INFORM TRAINING | 85 | 100 | 200 | 15.15 | 23.50 | QUERY PRO | 28422 | 15 |
| IN-G61 | INFORM TRAINING | 85 | 100 | 200 | 15.15 | 23.50 | QUERY PRO | 36576 | 50 |
| IN-J17 | INFORM CONCEPTS | 197 | 100 | 200 | 15.15 | 23.50 | QUERY PRO | 39531 | 5 |
| IN-J17 | INFORM CONCEPTS | 197 | 100 | 200 | 15.15 | 23.50 | QUERY PRO | 86071 | 10 |

Figure 9-3 Inventory Check List Report (Page 1 of 2)

```
IN-M41   INFORM MESSAGES        62    75    150    20.00    28.95  QUERY PRO       21116      10
IN-M41   INFORM MESSAGES        62    75    150    20.00    28.95  QUERY PRO       31012      35
IN-N19C  INFORM INSTALL CICS    11    75    150    11.30    18.95  QUERY PRO       53227      20
IN-N19C  INFORM INSTALL CICS    11    75    150    11.30    18.95  QUERY PRO       89003      35
IN-N19I  INFORM INSTALL IMS     38    25     50    11.30    18.95  QUERY PRO       53226       9
TR-M52   TRANSACT CONCEPTS      28    25     25    14.50    19.95  ONLINE WIZARD       -       -
TR-M55   TRANSACT INSTALL        3    25     25     9.00    15.95  ONLINE WIZARD   70652      12
TR-M89   TRANSACT MESSAGES      19    25     50    37.75    47.50  ONLINE WIZARD   72568      13
TR-M89   TRANSACT MESSAGES      19    25     50    37.75    47.50  ONLINE WIZARD   79031       5
TR-N94C  TRANSACT CICS GUIDE    30    25     25    23.25    39.95  ONLINE WIZARD       -       -
TR-N94I  TRANSACT IMS GUIDE     15    25     25    23.25    39.95  ONLINE WIZARD   21455       3
TR-O13   TRANSACT QUICK REF     59    25     50     6.25     7.95  ONLINE WIZARD       -       -
TR-O17   TRANSACT INST UPDT     25    25     25    16.40    23.95  ONLINE WIZARD       -       -
```

Figure 9-3 Inventory Check List Report (Page 2 of 2)

**Note:** For a complete version of this example, see Inventory Check List Report on page D-77.

Reports are created easily using the Report command group. The FORMAT command specifies page layout and global report characteristics (for example, date and page number). The COLUMNS command lists the fields to be reported. The TITLE command specifies title lines for the report.

Figure 9-4 shows all the commands needed to produce the Inventory Check List report.

- The first three statements comprise the CONTROL command group and specify application-related parameters and controls.
- The Report command group that follows includes all the commands that are used to specify report output.
- INVCHK1 is the label of the Report command group.
- REPORT identifies the type of command group.
- The fields to be reported are listed after the COLUMNS keyword.
- FORMAT is used here to specify the date in mm/dd/yy format.
- TITLE specifies the desired report title lines.

```
CONTROL  LISTGEN
FILE MASTER INPUT, NAME ITEMFILE
FILE REPORT
;
INVCHK1: REPORT,
         COLUMNS ITEMNO, ITEMNAME, QANONHND, REORDPT, ORDQTY,
               UNITCOST, ITMPRICE, VENDOR, ORDERNO, QTYBKORD
         FORMAT DATEFMT TODAY
         TITLE 'INVENTORY CHECKLIST'
END REPORT
```

Figure 9-4 Report Command Group Producing Inventory Checklist Report

**Note:** For a complete version of this example, see Inventory Check List Report on page D-77.

See the *VISION:Builder ASL Reference Guide* for descriptions of all commands and keywords.

For a moment, examine the report in Figure 9-3. In this example, VISION:Builder used defaults to control the majority of the formatting of the report. Notice that not all report selections were explicitly defined. For example, the report is printed on paper that is a fairly common size, yet the height and width of the report were not specified.

The first line of the report contains the date, report title, and a page number. The format of the date was selected but not the location. A title was specified but not which line of the report it should appear on or whether it should be centered on the line. The subject of page numbers was ignored. All of these defaults can be overridden using FORMAT command keywords.

The two lines of column headings are printed between two lines of dashes. By not making any specifications, the report defaulted to the installation standard heading separator character (-) and the column headings built into the file definition of the master file.

In the body of the report, the fields are in the order specified (left to right) on the COLUMNS command. However, there are some hidden specifications here: two blank spaces between report columns and commas inserted between the thousand's and hundred's digit, decimal points, and dollar signs. Furthermore, VISION:Builder determined the sequence of lines. When there is no other line sequence selection, VISION:Builder maintains the order of input, in this case, the order of the master file records.

The two blank spaces between report columns is a VISION:Builder standard that can be changed. The quantity fields are defined as numeric and the standard print edit calls for commas as used in normal numeric representation. The cost and price fields are defined as numeric with two decimal places, so VISION:Builder prints them accordingly. The dollar signs, however, are specified by you in the file definition as an output edit that applies only to the first occurrence of a field on a report page. You can override both of these edit specifications.

To review, the specified report requests are as follows:

- Date on each report page.
- Fields to be reported from the master file.
- Title for the report.

The file definition influenced the report in the following manner:

- Column headings.
- Print format of fields.
- Output edits causing the appearance of additional characters (for example, decimals, and dollar signs).

Many of the report specifications were VISION:Builder default standards:

- Location of date, title, page numbers.
- Centering of title lines on the page.
- Centering of column headings.
- Height and width of the page.
- Character for the column heading separator.
- Sequence of lines on the report.
- Spaces between report columns.
- Print format of report fields.

The rest of this section will review other reports that use different selections and reporting options. Most of the reports illustrate the effect of one or two report format options.

Figure 9-5 shows a sample ordering status report. This example shows how to:

- Display the date in a different format of MMM DD, YYYY rather than MM/DD/YY.
- Vary the spacing between certain report columns.

JUN 14, 2002 ORDERING REPORT PAGE 1

| ITEM NUMBER | ITEM NAME | QTY ON HAND | REORDER POINT | ORDER QTY |
|---|---|---|---|---|
| BL-B01 | BLDR QUICK REF | 249 | 100 | 200 |
| BL-E68 | BUILDER CONCEPTS | 87 | 100 | 200 |
| BL-F43 | BLDR MESSAGES | 16 | 100 | 200 |
| BL-H11A | BLDR HANDBOOK ASL | 56 | 100 | 200 |
| BL-H11F | BLDR HANDBOOK FIXED | 50 | 50 | 100 |
| BL-T01B | BLDR TRAINING BASIC | 15 | 5 | 5 |
| BL-T02A | BLDR TRAINING ADV | 15 | 5 | 5 |
| GN-N34 | DATA BASE CONCEPTS | 14 | 50 | 50 |
| IN-E13 | INFORM USER GUIDE | 175 | 100 | 200 |
| IN-G61 | INFORM TRAINING | 85 | 100 | 200 |
| IN-J17 | INFORM CONCEPTS | 197 | 100 | 200 |
| IN-M41 | INFORM MESSAGES | 62 | 75 | 150 |
| IN-N19C | INFORM INSTALL CICS | 11 | 75 | 150 |
| IN-N19I | INFORM INSTALL IMS | 38 | 25 | 50 |
| TR-M52 | TRANSACT CONCEPTS | 28 | 25 | 25 |
| TR-M55 | TRANSACT INSTALL | 3 | 25 | 25 |
| TR-M89 | TRANSACT MESSAGES | 19 | 25 | 50 |
| TR-N94C | TRANSACT CICS GUIDE | 30 | 25 | 25 |
| TR-N94I | TRANSACT IMS GUIDE | 15 | 25 | 25 |
| TR-O13 | TRANSACT QUICK REF | 59 | 25 | 50 |
| TR-O17 | TRANSACT INST UPDT | 25 | 25 | 25 |

Figure 9-5 Ordering Report

**Note:** For a complete version of this example, see Inventory Check List Report on page D-77.

Figure 9-6 shows the Report command group that generated the Ordering report. There are only two formatting differences between this format and the inventory check list format.

- The use of the value DATE rather than TODAY for the DATEFMT entry on the FORMAT command (see Figure 9-4), causing the date to be printed as JUN 14, 2002 rather than 06/14/02.
- The spacing between columns. By specifying SPACES 10 on the ITEM commands for both the QANONHND and ORDQTY fields, extra room was created between the two report columns, thereby adding 8 spaces to the normal two spaces that occur when no specification is made.

```
ORDERPT: REPORT,
         COLUMNS ITEMNO, ITEMNAME, QANONHND, REORDPT, ORDQTY
           FORMAT DATEFMT DATE
           ITEM QANONHND SPACES 10
           ITEM ORDQTY   SPACES 10
         TITLE 'ORDERING REPORT'
END REPORT
```

Figure 9-6 Report Command Group Producing Ordering Report

**Note:** For a complete version of this example, see Inventory Check List Report on page D-77.

Figure 9-7 shows a sample inventory count form with much of the format customized. This example shows you how to:

- Print two report images on the same page (a multiple image report).
- Print a portion of the item name by partial fielding (PF) ITEMNAME.
- Change the height and width of the report page.
- Eliminate the printing of column heading border.
- Double-space the report.

Alternate formats for the report title on a multiple image report are also described.

INVENTORY COUNT FORM PAGE 1

| ITEM NUMBER | ITEM NAME | IN STOCK | REORDER POINT | ORDER QTY |
|---|---|---|---|---|
| BL-B01 | BLDR QUICK REF | _____ | 100 | 200 |
| BL-E68 | BUILDER CONCEPT | _____ | 100 | 200 |
| BL-F43 | BLDR MESSAGES | _____ | 100 | 200 |
| BL-H11A | BLDR HANDBOOK A | _____ | 100 | 200 |
| BL-H11F | BLDR HANDBOOK F | _____ | 50 | 100 |
| BL-T01B | BLDR TRAINING B | _____ | 5 | 5 |
| BL-T02A | BLDR TRAINING A | _____ | 5 | 5 |

INVENTORY COUNT FORM PAGE 2

| ITEM NUMBER | ITEM NAME | IN STOCK | REORDER POINT | ORDER QTY |
|---|---|---|---|---|
| IN-N19C | INFORM INSTALL | _____ | 75 | 150 |
| IN-N19I | INFORM INSTALL | _____ | 25 | 50 |
| TR-M52 | TRANSACT CONCEP | _____ | 25 | 25 |
| TR-M55 | TRANSACT INSTAL | _____ | 25 | 25 |
| TR-M89 | TRANSACT MESSAG | _____ | 25 | 50 |
| TR-N94C | TRANSACT CICS G | _____ | 25 | 25 |
| TR-N94I | TRANSACT IMS GU | _____ | 25 | 25 |

Figure 9-7 Inventory Count Form (Page 1 of 2)

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| GN-N34 | DATA BASE CONCE | _____ | 50 | 50 | TR-O13 | TRANSACT QUICK | _____ | 25 | 50 |
| IN-E13 | INFORM USER GUI | _____ | 100 | 200 | TR-O17 | TRANSACT INST U | _____ | 25 | 25 |
| IN-G61 | INFORM TRAINING | _____ | 100 | 200 | | | | | |
| IN-J17 | INFORM CONCEPTS | _____ | 100 | 200 | | | | | |
| IN-M41 | INFORM MESSAGES | _____ | 75 | 150 | | | | | |

Figure 9-7 Inventory Count Form (Page 2 of 2)

**Note:** For a complete version of this example, see Inventory Check List Report on page D-77.

Figure 9-8 shows the Report command group. In this case, the report title prints in the default position. The title INVENTORY COUNT FORM, along with the page number, appears over each report image on the report page.

As an alternative, this report can also be output with a single title and page number positioned across the report page. To do this, specify IMGTITLE PHYPAGE on the FORMAT command. To fit special binders, this report uses only 35 lines per page, specified by HEIGHT 35 on the FORMAT command.

To keep the number of pages to a minimum and to make the report easy to read, the report is double-spaced and two report images are printed on one sheet of paper.

You can use double-spacing by entering DETAILSPACING on the FORMAT command (see Figure 9-8). To get two report pages on the same sheet, report width is specified as 35 characters on the WIDTH keyword and the IMAGES keyword is specified as 2 (two report images per page).

```
REPORT COLUMNS ITEMNO, PF(ITEMNAME,1,15), T.INSTOCK, REORDPT, ORDQTY
       FORMAT BORDER NO, DETAILSPACING 2,
              IMAGES 2,   WIDTH 55,   HEIGHT 35
       TITLE 'INVENTORY COUNT FORM'
END REPORT
```

Figure 9-8 Report Command Group: Inventory Count Form

**Note:** For a complete version of this example, see Inventory Check List Report on page D-77.

The COLUMNS keyword on the REPORT command shows each field in order as it appears on the report. Because only the first 15 characters of the item name are to be printed, the ITEMNAME is specified using the built-in PF (partial fielding) function. It defines which portion of the character field is to be printed. In this case, PF(ITEMNAME,1,15) prints ITEMNAME starting with the first character for a length of 15. For more information about partial fielding, see Chapter 4, *Field Characteristics and Handling*.

The temporary field requires a qualified field name (T.INSTOCK) to indicate the origin of the field. Otherwise, temporary fields are reported just like any other field. They can contain values computed by the application or, as in this example, a constant value. This temporary field contains five underscores (_____).

Finally, BORDER NO was specified on the FORMAT command. By suppressing the column heading border, two lines per report page were saved, thereby adding at least one line of data per page.

See the *VISION:Builder ASL Reference Guide* for more information about the rules for the column heading character entry:

# Sequencing and Grouping Report Records

Early in this chapter, the sequence of the lines on the report was determined by the sequence of the records on the master file. This section will detail how that sequence can be controlled and how to use the sequence to change the appearance of the report.

## Controlling the Sequence of Report Records

VISION:Builder allows up to nine fields to be used as sort control fields in sequencing report lines. These fields are identified on the ORDER command in the Report command group.

**Note:** You can provide multiple ORDER commands. The first field specified on the first ORDER command is the most major sort field. Subsequent fields become increasingly minor.

## Grouping Report Data

Grouping data makes a report easier to read. The two major actions used to group data are sorting and control breaks. A control break occurs when the value of a field changes from the value in the preceding record. Using a control break on the Group command, you can group common data and insert totals and subtotals.

VISION:Builder provides a variety of functions that can be initiated at control breaks specified on the Group command. At an identified control break field, VISION:Builder prints the first occurrence of a specific value in the field and leaves that column blank until a new value occurs or a new page is started. This improves the readability of a report by suppressing the printing of the same value over and over again. When you specify a level 1 control break, VISION:Builder skips a line before printing the new value of the control field. If skipping a line is not wanted, level 2 can be specified as the first control break.

In order to take maximum advantage of this feature, it is necessary to sort the report by the values of control break fields. This ensures that proper grouping occurs.

Figure 9-9 shows the Report command group with two additional entries for the vendor number. The vendor has been designated as the primary sort control and as the primary control break.

```
REPORT  ITEMNO, ITEMNAME, VENDOR, QANONHND, ORDQTY,
        UNITCOST, T.COSTVALU, ITMPRICE, T.SALEVALU
   FORMAT DATEFMT DATE
   TITLE 'STOCK COST AND PRICE VALUE COMPARISON'
   ORDER BY VENDOR
   GROUP BY VENDOR
END REPORT
```

Figure 9-9 Report Command Group: Sort Sequence and Control Breaks

**Note:** For a complete version of this example, see Inventory Check List Report on page D-77.

In the report shown in Figure 9-10, note that the results of specifying VENDOR as a control break field and a sort sequence field. All documents ordered from each vendor are grouped together. The vendor number is printed only on the first item line and there is a blank line between vendors. Note that the vendor is printed once for each sorted group.

JUN 14, 2002 STOCK COST AND PRICE VALUE COMPARISON PAGE 1

| ITEM NUMBER | ITEM NAME | VENDOR | QTY ON HAND | ORDER QTY | UNIT COST | VALUE AT COST | ITEM PRICE | VALUE AT SALE PRICE |
|---|---|---|---|---|---|---|---|---|
| BL-B01 | BLDR QUICK REF | ACCESS MASTERS | 249 | 200 | $7.00 | $1,743.00 | $7.95 | $1,979.55 |
| BL-E68 | BUILDER CONCEPTS | | 87 | 200 | 20.25 | 1,761.75 | 27.50 | 2,392.50 |
| BL-F43 | BLDR MESSAGES | | 16 | 200 | 12.25 | 196.00 | 21.00 | 336.00 |
| BL-H11A | BLDR HANDBOOK ASL | | 56 | 200 | 23.25 | 1,302.00 | 31.00 | 1,736.00 |
| BL-H11F | BLDR HANDBOOK FIXED | | 50 | 100 | 23.75 | 1,187.50 | 35.00 | 1,750.00 |
| BL-T01B | BLDR TRAINING BASIC | IN HOUSE | 15 | 5 | $.00 | $.00 | $775.00 | $11,625.00 |
| BL-T02A | BLDR TRAINING ADV | | 15 | 5 | .00 | .00 | 1,000.00 | 15,000.00 |
| TR-M52 | TRANSACT CONCEPTS | ONLINE WIZARD | 28 | 25 | $14.50 | $406.00 | $19.95 | $558.60 |
| TR-M55 | TRANSACT INSTALL | | 3 | 25 | 9.00 | 27.00 | 15.95 | 47.85 |
| TR-M89 | TRANSACT MESSAGES | | 19 | 50 | 37.75 | 717.25 | 47.50 | 902.50 |
| TR-N94C | TRANSACT CICS GUIDE | | 30 | 25 | 23.25 | 697.50 | 39.95 | 1,198.50 |
| TR-N94I | TRANSACT IMS GUIDE | | 15 | 25 | 23.25 | 348.75 | 39.95 | 599.25 |
| TR-O13 | TRANSACT QUICK REF | | 59 | 50 | 6.25 | 368.75 | 7.95 | 469.05 |
| TR-O17 | TRANSACT INST UPDT | | 25 | 25 | 16.40 | 410.00 | 23.95 | 598.75 |
| IN-E13 | INFORM USER GUIDE | QUERY PRO | 175 | 200 | $23.25 | $4,068.75 | $28.95 | $5,066.25 |
| IN-G61 | INFORM TRAINING | | 85 | 200 | 15.15 | 1,287.75 | 23.50 | 1,997.50 |
| IN-J17 | INFORM CONCEPTS | | 197 | 200 | 15.15 | 2,984.55 | 23.50 | 4,629.50 |
| IN-M41 | INFORM MESSAGES | | 62 | 150 | 20.00 | 1,240.00 | 28.95 | 1,794.90 |
| IN-N19C | INFORM INSTALL CICS | | 11 | 150 | 11.30 | 124.30 | 18.95 | 208.45 |
| IN-N19I | INFORM INSTALL IMS | | 38 | 50 | 11.30 | 429.40 | 18.95 | 720.10 |
| GN-N34 | DATA BASE CONCEPTS | TCOM CONSULTING | 14 | 50 | $33.25 | $465.50 | $39.95 | $559.30 |

Figure 9-10 Resulting Report: Sort Sequence and Control Breaks

**Note:** For a complete version of this example, see Inventory Check List Report on page D-77.

## Levels of Control Break

A control break field is specified on the GROUP command. Further group qualifications may be controlled by entering a number (1 through 9) on the AT LEVEL keyword of the GROUP command. This number is called the control break level. The larger the number given to the control break, the more minor the level.

Up to nine control breaks can be specified. A change in value of a field at one level causes a break at all more minor levels, regardless of the contents of those fields at that time. For example, if you specified three control break fields and numbered them 1, 2, and 4, a change in the value of control break field 2 causes all control break action to be performed for control levels 2 and 4, but not for level 1.

See the *VISION:Builder ASL Reference Guide* for more information.

The next report shows the use of two levels of control break. The purpose is to show the total number of documents back ordered (QTYBKORD) for each item (ITEMNO) and the total number of documents back ordered for each vendor (VENDOR). Figure 9-11 shows the Report command group and Figure 9-12 shows the report.

```
REPORT VENDOR ITEMNO ITEMNAME ORDERNO QTYBKORD
    FORMAT DATEFMT DATE
    TITLE 'BACKORDERS BY VENDOR AND ITEM'
  ORDER BY VENDOR ITEMNO
  GROUP BY VENDOR
  GROUP BY ITEMNO
  TOTAL QTYBKORD BY ITEMNO
END REPORT
```

Figure 9-11 Report Command Group: Multiple Control Breaks

**Note:** For a complete version of this example, see Inventory Check List Report on page D-77.

```
JUN 14, 2002                                  BACKORDERS BY VENDOR AND ITEM            PAGE 1

-----------------------------------------------------------------------------------------------
                  VENDOR        ITEM            ITEM          ORDER   QTY BACK
                                NUMBER          NAME          NUMBER  ORDERED
-----------------------------------------------------------------------------------------------
              ACCESS MASTERS   BL-E68   BUILDER CONCEPTS      45097       90

ITEMNO   TOTAL ACCESS MASTERS  BL-E68                                     90

                               BL-F43   BLDR MESSAGES         10081       50
                                        BLDR MESSAGES         31565       50

ITEMNO   TOTAL ACCESS MASTERS  BL-F43                                    100

                               BL-H11A  BLDR HANDBOOK ASL     10987       25
                                        BLDR HANDBOOK ASL     12568       50

ITEMNO   TOTAL ACCESS MASTERS  BL-H11A                                    75

                               BL-H11F  BLDR HANDBOOK FIXED   65012       60

ITEMNO   TOTAL ACCESS MASTERS  BL-H11F                                    60

VENDOR   TOTAL ACCESS MASTERS                                            325

              IN HOUSE         BL-T01B  BLDR TRAINING BASIC   90311        2

ITEMNO   TOTAL IN HOUSE        BL-T01B                                     2

VENDOR   TOTAL IN HOUSE                                                    2
```

Figure 9-12 Resulting Report: Sort Sequence and Multiple Control Breaks

**Note:** For a complete version of this example, see Inventory Check List Report on page D-77.

The Report command group shows the report data being sorted into item number sequence within vendor. Both of those fields are also specified as control breaks, or grouping fields. In addition, totals are printed for backorder quantities when the item number changes. Since ITEMNO is specified on the second GROUP command, this is considered to be control break level 2.

The quantities of documents back ordered are added as the data is reported in the specified sequence. When the value in either control break field changes, the total of the back ordered documents accumulated since the last change in that field is printed. If the change is at level 2 (if the item number has changed), only the total for the order is printed. If the change is at level 1 (if the vendor changes), both totals are printed and a line is skipped. As mentioned earlier, a change in a control break field causes control break actions to take place for the current level and more minor levels.

In most situations, a change at a more major control break level implies a break at all more minor levels. In this example, continuing the total for an item when the vendor changes would not make sense even if the item numbers were the same because the purpose is to find totals for each item for a given vendor.

## Summarizing Report Data

VISION:Builder provides eight summaries for you to apply to fields. These are total, cumulative, count, maximum, minimum, average, percent, and ratio. Summaries are printed whenever a control break field changes value or at the end of a report. Control break fields cannot be summarized.

In the next example, after grouping the various documents by vendor, totals of the stock value at both cost and sale price are printed. The SUMMARYONLY entry on the REPORT statement allows you to specify one of eight summary types. Use the TOTAL command to specify that a total is to be taken at any level of control break. The choice is limited in this example because there is only one control break. The Report command group is shown in Figure 9-13 and the resulting report in Figure 9-14.

You may want to specify more than one type of summary on the same field. For example, you may want both a total and an average computed for the ORDQTY field. Specifying more than one type of summary is described in Reports Using Summaries Only on page 9-14.

```
REPORT COLUMNS  ITEMNO, ITEMNAME, VENDOR, QANONHND, ORDQTY,
              UNITCOST, T.COSTVALU, ITMPRICE, T.SALEVALU
   FORMAT DATEFMT DATE
   TITLE 'STOCK COST AND PRICE VALUE COMPARISON'
   ORDER BY VENDOR
   GROUP BY VENDOR
     TOTAL T.COSTVALU BY VENDOR
     TOTAL T.SALEVALU BY VENDOR
END REPORT
```

Figure 9-13 Report Command Group: Summary Report with Totals

**Note:** For a complete version of this example, see Inventory Check List Report on page D-77.

JUN 14, 2002 STOCK COST AND PRICE VALUE COMPARISON PAGE 1

| | | ITEM NUMBER | ITEM NAME | VENDOR | QTY ON HAND | ORDER QTY | UNIT COST | VALUE AT COST | ITEM PRICE | VALUE AT SALE PRICE |
|---|---|---|---|---|---|---|---|---|---|---|
| | | BL-B01 | BLDR QUICK REF | ACCESS MASTERS | 249 | 200 | $7.00 | $1,743.00 | $7.95 | $1,979.55 |
| | | BL-E68 | BUILDER CONCEPTS | | 87 | 200 | 20.25 | 1,761.75 | 27.50 | 2,392.50 |
| | | BL-F43 | BLDR MESSAGES | | 16 | 200 | 12.25 | 196.00 | 21.00 | 336.00 |
| | | BL-H11A | BLDR HANDBOOK ASL | | 56 | 200 | 23.25 | 1,302.00 | 31.00 | 1,736.00 |
| | | BL-H11F | BLDR HANDBOOK FIXED | | 50 | 100 | 23.75 | 1,187.50 | 35.00 | 1,750.00 |
| VENDOR | TOTAL | | | ACCESS MASTERS | | | | $6,190.25 | | $8,194.05 |
| | | BL-T01B | BLDR TRAINING BASIC | IN HOUSE | 15 | 5 | $.00 | $.00 | $775.00 | $11,625.00 |
| | | BL-T02A | BLDR TRAINING ADV | | 15 | 5 | .00 | .00 | 1,000.00 | 15,000.00 |
| VENDOR | TOTAL | | | IN HOUSE | | | | $.00 | | $26,625.00 |
| | | TR-M52 | TRANSACT CONCEPTS | ONLINE WIZARD | 28 | 25 | $14.50 | $406.00 | $19.95 | $558.60 |
| | | TR-M55 | TRANSACT INSTALL | | 3 | 25 | 9.00 | 27.00 | 15.95 | 47.85 |
| | | TR-M89 | TRANSACT MESSAGES | | 19 | 50 | 37.75 | 717.25 | 47.50 | 902.50 |
| | | TR-N94C | TRANSACT CICS GUIDE | | 30 | 25 | 23.25 | 697.50 | 39.95 | 1,198.50 |
| | | TR-N94I | TRANSACT IMS GUIDE | | 15 | 25 | 23.25 | 348.75 | 39.95 | 599.25 |
| | | TR-O13 | TRANSACT QUICK REF | | 59 | 50 | 6.25 | 368.75 | 7.95 | 469.05 |
| | | TR-O17 | TRANSACT INST UPDT | | 25 | 25 | 16.40 | 410.00 | 23.95 | 598.75 |
| VENDOR | TOTAL | | | ONLINE WIZARD | | | | $2,975.25 | | $4,374.50 |
| | | IN-E13 | INFORM USER GUIDE | QUERY PRO | 175 | 200 | $23.25 | $4,068.75 | $28.95 | $5,066.25 |
| | | IN-G61 | INFORM TRAINING | | 85 | 200 | 15.15 | 1,287.75 | 23.50 | 1,997.50 |
| | | IN-J17 | INFORM CONCEPTS | | 197 | 200 | 15.15 | 2,984.55 | 23.50 | 4,629.50 |
| | | IN-M41 | INFORM MESSAGES | | 62 | 150 | 20.00 | 1,240.00 | 28.95 | 1,794.90 |
| | | IN-N19C | INFORM INSTALL CICS | | 11 | 150 | 11.30 | 124.30 | 18.95 | 208.45 |
| | | IN-N19I | INFORM INSTALL IMS | | 38 | 50 | 11.30 | 429.40 | 18.95 | 720.10 |
| VENDOR | TOTAL | | | QUERY PRO | | | | $10,134.75 | | $14,416.70 |
| | | GN-N34 | DATA BASE CONCEPTS | TCOM CONSULTING | 14 | 50 | $33.25 | $465.50 | $39.95 | $559.30 |
| VENDOR | TOTAL | | | TCOM CONSULTING | | | | $465.50 | | $559.30 |

Figure 9-14 Summary Report with Totals

**Note:** For a complete version of this example, see Inventory Check List Report on page D-77.

This report is only a first step. The rough format is difficult to read. Too much of the report formatting task was left to defaults.

Unless otherwise specified, VISION:Builder reserves the first 14 report line positions for summary labels (such as a label for a calculated total line). When the summary label combined with the number of fields makes the report line longer than the paper, VISION:Builder "folds" the line (that is, takes what falls off the right side of the report and makes a second line out of it). The same thing would happen to the column headings and this would make the report even more difficult to read.

# Creating Subtitles and Adjusting Summary Labels

When a report is long, the use of subtitles can help you to quickly scan through the report to find the information you need. A subtitle prints below the main title of the report and consists of the contents of the fields specified as SUBTITLE fields (on the GROUP command).

To make this report even easier to read, each vendor starts on a new page. The vendor and product code print as subtitles and grand totals accumulate for all vendors.

The vendor and product code subtitles are obtained by specifying NEWPAGE (page subtitle) on the GROUP command (see Figure 9-15). A page subtitle causes every new value of the specified field to appear as a subtitle on a new page.

To get the product code to print under the vendor, ENDLINE is specified on the ITEM command for the VENDOR field (see Figure 9-15). This entry tells VISION:Builder to start a new line with the next field in the COLUMNS list on the REPORT command.

```
REPORT GRANDSUMS,
       COLUMNS  VENDOR, T.PRODCODE, ITEMNO, ITEMNAME, QANONHND,
                ORDQTY, UNITCOST, T.COSTVALU, ITMPRICE, T.SALEVALU
   FORMAT DATEFMT DATE
   TITLE 'STOCK COST AND PRICE VALUE COMPARISON'
   ORDER BY VENDOR
   GROUP BY VENDOR T.PRODCODE SUBTITLE NEWPAGE
   ITEM  VENDOR ENDLINE
     TOTAL T.COSTVALU BY VENDOR
     TOTAL T.SALEVALU BY VENDOR
END REPORT
```

Figure 9-15 Report Command Group: Summary Report with Subtitles

**Note:** For a complete version of this example, see Inventory Check List Report on page D-77.

On the REPORT command, the GRANDSUMS keyword entry produces the grand totals. This entry tells VISION:Builder to keep a running total of fields being summarized and to add a page presenting that information at the end of the report.

SUMMARYLABELS NOSPACE on the FORMAT command indicates that columns of data begin at the left margin of the report. You are responsible for any interference between summaries and summary labels on the total lines. After moving the vendor to a subtitle line, the first two fields in the detail line are the item number and the item name. Because neither is totaled, there is room for the summary labels without a conflict.

As an alternative, you can enter SUMMARYLABELS SUPPRESS on the FORMAT command to suppress summary labels completely.

Figure 9-16 shows the new report layout.

**Note:** There is an implied control break for the first line of a report so the first page contains the proper subtitles.

The subtitle entries on the FORMAT command allows you to select repeating or non-repeating subtitles by report to override your M4PARAMS setting. For more information about M4PARAMS, refer to the *VISION:Builder for OS/390 Installation Guide* (OS/390 users) or the *VISION:Builder Installation Guide for VSE* (VSE users).

```
JUN 14, 2002                          STOCK COST AND PRICE VALUE COMPARISON                          PAGE 1

  ACCESS MASTERS
  BL

-------------------------------------------------------------------------------------------------------------
                ITEM          ITEM          QTY ON  ORDER   UNIT     VALUE       ITEM     VALUE AT
               NUMBER         NAME           HAND    QTY    COST    AT COST      PRICE    SALE PRICE
-------------------------------------------------------------------------------------------------------------
               BL-B01   BLDR QUICK REF        249    200     $7.00  $1,743.00     $7.95   $1,979.55
               BL-E68   BUILDER CONCEPTS       87    200     20.25   1,761.75     27.50    2,392.50
               BL-F43   BLDR MESSAGES          16    200     12.25     196.00     21.00      336.00
               BL-H11A  BLDR HANDBOOK ASL      56    200     23.25   1,302.00     31.00    1,736.00
               BL-H11F  BLDR HANDBOOK FIXED    50    100     23.75   1,187.50     35.00    1,750.00

VENDOR   TOTAL                                                      $6,190.25             $8,194.05


JUN 14, 2002                          STOCK COST AND PRICE VALUE COMPARISON                          PAGE 2

  IN HOUSE
  BL

-------------------------------------------------------------------------------------------------------------
                ITEM          ITEM          QTY ON  ORDER   UNIT     VALUE       ITEM     VALUE AT
               NUMBER         NAME           HAND    QTY    COST    AT COST      PRICE    SALE PRICE
-------------------------------------------------------------------------------------------------------------
               BL-T01B  BLDR TRAINING BASIC    15      5      $.00       $.00   $775.00  $11,625.00
               BL-T02A  BLDR TRAINING ADV      15      5       .00        .00  1,000.00   15,000.00

VENDOR   TOTAL                                                          $.00            $26,625.00
.
.
.
JUN 14, 2002                          STOCK COST AND PRICE VALUE COMPARISON                          PAGE 6


-------------------------------------------------------------------------------------------------------------
                ITEM          ITEM          QTY ON  ORDER   UNIT     VALUE       ITEM     VALUE AT
               NUMBER         NAME           HAND    QTY    COST    AT COST      PRICE    SALE PRICE
-------------------------------------------------------------------------------------------------------------
GRAND    TOTAL                                                     $19,765.75            $54,169.55
```

Figure 9-16 Summary Report with Totals

**Note:** For a complete version of this example, see Inventory Check List Report on page D-77.

# Reports Using Summaries Only

With VISION:Builder, limiting the contents of a report to the summary lines is a simple request. For example, you are asked to prepare a report showing only the totals from the report in the previous example (see Figure 9-16). With most programming languages, you would have to sort the master file to change it from item number sequence to vendor number sequence or set up internal summary buckets to accumulate the totals as the data is read. VISION:Builder makes it easy; request the totals in the usual fashion and just ask for a SUMMARY_ONLY on the REPORT command.

Figure 9-17 shows the previous Report command group modified to produce a summary only report. In addition, changes have been made to print grand summaries on the same report page rather than on a separate report page. The report is shown in Figure 9-19.

```
REPORT SUMMARY_ONLY SINGLESPACE,
       COLUMNS  VENDOR, T.PRODCODE, T.COSTVALU, T.SALEVALU
   FORMAT DATEFMT DATE
   TITLE 'SUMMARY OF COST AND PRICE VALUE COMPARISON'
   ORDER BY VENDOR
   GROUP BY VENDOR T.PRODCODE
     TOTAL    T.COSTVALU BY VENDOR
     CUMULATE T.COSTVALU AT G
     TOTAL    T.SALEVALU BY VENDOR
     CUMULATE T.SALEVALU AT G
END REPORT
```

Figure 9-17 Report Command Group: Summary Only Report

**Note:** For a complete version of this example, see on page D-77.

Most of the fields listed on the REPORT command (see Figure 9-17) are deleted because those detail fields do not print on the summary line. The SINGLESPACE entry on the REPORT command (see Figure 9-17) indicates a summary only Report with single-spacing of the summary lines and double-spacing after the level one summary lines. Because there is only a level one control break, the report appears double-spaced.

Had there been more than one summary level, the lines would have been single-spaced with an extra line following the level one summary. The default is double-spacing between all summary lines for summary-only reports.

Had the GRANDSUMS keyword been used for these reports, the grand totals would be printed on a separate page. Instead, a grand cumulative summary is specified by placing the CUMULATE command for each field and specifying the grand level AT G so that the grand summaries print on the same report page (see Figure 9-17). This is also an example of how to request more than one type of summary on the same field.

This same example is shown in Figure 9-18 with one modification that eliminates the summary labels.

The one change made to the Report command group (see Figure 9-18) is the addition of SUMMARYLABELS SUPPRESS on the FORMAT command.

```
REPORT SUMMARY_ONLY SINGLESPACE,
       COLUMNS  VENDOR, T.PRODCODE, T.COSTVALU, T.SALEVALU
   FORMAT DATEFMT DATE SUMMARYLABELS SUPPRESS
   TITLE 'SUMMARY OF COST AND PRICE VALUE COMPARISON'
   ORDER BY VENDOR
   GROUP BY VENDOR T.PRODCODE
     TOTAL    T.COSTVALU BY VENDOR
     CUMULATE T.COSTVALU AT G
     TOTAL    T.SALEVALU BY VENDOR
     CUMULATE T.SALEVALU AT G
END REPORT
```

Figure 9-18 Report Command Group: Summary Only Report without Labels

**Note:** For a complete version of this example, see Inventory Check List Report on page D-77.

Figure 9-19 shows the report with summary labels and Figure 9-20 shows the report without the summary labels.

```
JUN 14, 2002                          SUMMARY OF COST AND PRICE VALUE COMPARISON                PAGE 1

-----------------------------------------------------------------------------------------------------------
                   VENDOR       PRODCODE    VALUE      VALUE AT
                                            AT COST    SALE PRICE
-----------------------------------------------------------------------------------------------------------
VENDOR   TOTAL  ACCESS MASTERS      BL      $6,190.25   $8,194.05

VENDOR   TOTAL  IN HOUSE            BL           $.00  $26,625.00

VENDOR   TOTAL  ONLINE WIZARD       TR      $2,975.25   $4,374.50

VENDOR   TOTAL  QUERY PRO           IN     $10,134.75  $14,416.70

VENDOR   TOTAL  TCOM CONSULTING     GN        $465.50     $559.30

GRAND    CUM.                             $19,765.75  $54,169.55
```

Figure 9-19 Summary Only Report

**Note:** For a complete version of this example, see Inventory Check List Report on page D-77.

```
JUN 14, 2002                          SUMMARY OF COST AND PRICE VALUE COMPARISON                PAGE 1

-----------------------------------------------------------------------------------------------------------
      VENDOR        PRODCODE    VALUE      VALUE AT
                                AT COST    SALE PRICE
-----------------------------------------------------------------------------------------------------------
  ACCESS MASTERS      BL      $6,190.25   $8,194.05

  IN HOUSE            BL           $.00  $26,625.00

  ONLINE WIZARD       TR      $2,975.25   $4,374.50

  QUERY PRO           IN     $10,134.75  $14,416.70

  TCOM CONSULTING     GN        $465.50     $559.30

                            $19,765.75  $54,169.55
```

Figure 9-20 Summary Only Report without Labels

**Note:** For a complete version of this example, see Inventory Check List Report on page D-77.

# Field Output Edit

The following two reports and report definitions show various output formats. Figure 9-21 shows the Report command group with a picture specification, as well as a page control break and subtitle control breaks.

The ITEM command can be used with any field on a report; it has a large number of available options to tailor the format of both character and numeric fields.

In this example, the edit picture specifies the words PRODUCT GROUP will be followed by a space and the value of the control break field, which is simply a partial field of the first two characters of the item number. Any combination of data and literal characters can be mixed within the edit picture. The literals are written in the output edit field and individual characters of the data field are identified using a delimiter character.

The product group, or partial field of the item number, is also assigned a page control break. The control break causes the value of the field to be printed as a subtitle on a new page.

A subtitle control break is specified for the field VENDOR. The control break causes the field values to be printed on a separate line between groups of detail lines, preceded and followed by blank lines. An edit picture is specified for the field VENDOR. It specifies that the word VENDOR will be followed by a space and the value of the field VENDOR.

```
REPORT  PF(ITEMNO,1,2), VENDOR, ITEMNO, ITEMNAME,
        UNITCOST, T.NEWCOST, ITMPRICE, T.NEWPRICE
   ITEM PF(ITEMNO,1,2) PICTURE P'PRODUCT GROUP ##'
   ITEM VENDOR         PICTURE P'VENDOR ###############'
   FORMAT DATEFMT DATE
   TITLE 'RECOMMENDED NEW DOCUMENT PRICES'
   TITLE 'PRODUCT BY VENDOR AND ITEM'
   ORDER BY PF(ITEMNO,1,2) VENDOR ITEMNO ITEMNAME
   GROUP BY PF(ITEMNO,1,2) SUBTITLE NEWPAGE
   GROUP BY VENDOR         SUBTITLE
```

Figure 9-21 Report Command Group: Field Output Edit

**Note:** For a complete version of this example, see Inventory Check List Report on page D-77.

Figure 9-22 shows this report with an edited page subtitle and edited normal subtitle.

```
JUN 14, 2002                          RECOMMENDED NEW DOCUMENT PRICES                    PAGE 1
                                        PRODUCT BY VENDOR AND ITEM

  PRODUCT GROUP BL

-------------------------------------------------------------------------------------------------
   ITEM          ITEM          UNIT     EXPECTED     ITEM     RECOMMENDED
  NUMBER         NAME          COST     NEW COST     PRICE     NEW PRICE
-------------------------------------------------------------------------------------------------

  VENDOR ACCESS MASTERS

  BL-B01   BLDR QUICK REF        $7.00       $7.63      $7.95       $9.14
  BL-E68   BUILDER CONCEPTS      20.25       22.07      27.50       31.62
  BL-F43   BLDR MESSAGES         12.25       13.35      21.00       24.15
  BL-H11A  BLDR HANDBOOK ASL     23.25       25.34      31.00       35.65
  BL-H11F  BLDR HANDBOOK FIXED   23.75       25.88      35.00       40.25

  VENDOR IN HOUSE

  BL-T01B  BLDR TRAINING BASIC    $.00        $.00    $775.00     $891.25
  BL-T02A  BLDR TRAINING ADV       .00         .00  1,000.00   1,150.00


JUN 14, 2002                          RECOMMENDED NEW DOCUMENT PRICES                    PAGE 2
                                        PRODUCT BY VENDOR AND ITEM

  PRODUCT GROUP GN
```

Figure 9-22 Resulting Report with Field Output Edit (Page 1 of 2)

```
------------------------------------------------------------------------------------------------------------
  ITEM          ITEM          UNIT     EXPECTED     ITEM     RECOMMENDED
 NUMBER         NAME          COST     NEW COST     PRICE     NEW PRICE
------------------------------------------------------------------------------------------------------------

 VENDOR TCOM CONSULTING

 GN-N34   DATA BASE CONCEPTS     $33.25      $36.24     $39.95      $45.94
```

Figure 9-22 Resulting Report with Field Output Edit (Page 2 of 2)

**Note:** For a complete version of this example, see Inventory Check List Report on page D-77.

Figure 9-23 shows the Report command group using non-print fields. When non-print is specified, the field is available for sort control, control breaks, and summaries but does not appear on the detail line of the report.

```
REPORT  PF(ITEMNO,1,2), ITEMNO, ITEMNAME,
        T.NEWCOST, T.NEWPRICE, T.DIFCOST, T.DIFPRICE
   ITEM PF(ITEMNO,1,2) PICTURE P'PRODUCT GROUP ##'
   ITEM T.DIFCOST  NONPRINT PICTURE P'$$$,$$$,$$9.99' SPACES 5
   ITEM T.DIFPRICE NONPRINT PICTURE P'$$$,$$$,$$9.99' SPACES 5
   FORMAT DATEFMT DATE
   TITLE 'ESTIMATED COST AND PRICE'
   TITLE 'VALUES AND DIFFERENCES'
   ORDER BY PF(ITEMNO,1,2)
   GROUP BY PF(ITEMNO,1,2) SUBTITLE NEWPAGE
   CUMULATE T.DIFCOST  AT G
   CUMULATE T.DIFPRICE AT G
   TOTAL T.DIFCOST  BY PF(ITEMNO,1,2)
   TOTAL T.DIFPRICE BY PF(ITEMNO,1,2)
END REPORT
```

Figure 9-23 Report Command Group: Non-Print Fields

The report in Figure 9-24 shows the effect of the non-print control specification, level one totals, and grand totals on the report. Totals are requested but nothing is printed on the detail lines. Totals are printed at each control break and the grand total is printed after the final control break total.

```
UN 14, 2002                                ESTIMATED COST AND PRICE                          PAGE 1
                                            VALUES AND DIFFERENCES

 PRODUCT GROUP BL

------------------------------------------------------------------------------------------------------------
             ITEM          ITEM          EXPECTED   RECOMMENDED      ESTIMATED         ESTIMATED
            NUMBER         NAME          NEW COST    NEW PRICE       COST VALUE        PRICE VALUE
------------------------------------------------------------------------------------------------------------
             BL-B01   BLDR QUICK REF          $7.63       $9.14
             BL-E68   BUILDER CONCEPTS        22.07       31.62
             BL-F43   BLDR MESSAGES           13.35       24.15
             BL-H11A  BLDR HANDBOOK ASL       25.34       35.65
             BL-H11F  BLDR HANDBOOK FIXED     25.88       40.25
             BL-T01B  BLDR TRAINING BASIC       .00      891.25
             BL-T02A  BLDR TRAINING ADV         .00    1,150.00

TEMNO   TOTAL                                                          $7.77           $284.61

UN 14, 2002                                ESTIMATED COST AND PRICE                          PAGE 2
                                            VALUES AND DIFFERENCES

 PRODUCT GROUP GN

------------------------------------------------------------------------------------------------------------
             ITEM          ITEM          EXPECTED   RECOMMENDED      ESTIMATED         ESTIMATED
            NUMBER         NAME          NEW COST    NEW PRICE       COST VALUE        PRICE VALUE
------------------------------------------------------------------------------------------------------------
```

Figure 9-24 Resulting Report with Non-Print-Field (Page 1 of 2)

```
              GN-N34   DATA BASE CONCEPTS        $36.24      $45.94

TEMNO   TOTAL                                                                  $2.99            $5.99

UN 14, 2002                                    ESTIMATED COST AND PRICE                                PAGE 3
                                                VALUES AND DIFFERENCES

 PRODUCT GROUP IN

-----------------------------------------------------------------------------------------------------------
              ITEM            ITEM           EXPECTED   RECOMMENDED       ESTIMATED          ESTIMATED
             NUMBER           NAME           NEW COST    NEW PRICE        COST VALUE        PRICE VALUE
-----------------------------------------------------------------------------------------------------------
              IN-E13   INFORM USER GUIDE        $25.34      $33.29
              IN-G61   INFORM TRAINING           16.51       27.02
              IN-J17   INFORM CONCEPTS           16.51       27.02
              IN-M41   INFORM MESSAGES           21.80       33.29
              IN-N19C  INFORM INSTALL CICS       12.31       21.79
              IN-N19I  INFORM INSTALL IMS        12.31       21.79

TEMNO   TOTAL                                                                  $8.63           $21.40

UN 14, 2002                                    ESTIMATED COST AND PRICE                                PAGE 4
                                                VALUES AND DIFFERENCES

 PRODUCT GROUP TR

-----------------------------------------------------------------------------------------------------------
              ITEM            ITEM           EXPECTED   RECOMMENDED       ESTIMATED          ESTIMATED
             NUMBER           NAME           NEW COST    NEW PRICE        COST VALUE        PRICE VALUE
-----------------------------------------------------------------------------------------------------------
              TR-M52   TRANSACT CONCEPTS        $15.80      $22.94
              TR-M55   TRANSACT INSTALL           9.81       18.34
              TR-M89   TRANSACT MESSAGES         41.14       54.62
              TR-N94C  TRANSACT CICS GUIDE       25.34       45.94
              TR-N94I  TRANSACT IMS GUIDE        25.34       45.94
              TR-O13   TRANSACT QUICK REF         6.81        9.14
              TR-O17   TRANSACT INST UPDT        17.87       27.54

TEMNO   TOTAL                                                                 $11.71           $29.26
RAND    CUM.                                                                  $31.10          $341.26
```

Figure 9-24 Resulting Report with Non-Print-Field (Page 2 of 2)

# Reporting Optimization

VISION:Builder provides you with the capability to optimize report processing by specifying the appropriate trade-off between memory utilization and I/O or CPU time for an application. You can optimize report processing using single-step processing (SSP)

Single-step processing (SSP) allows you to execute the three phases of VISION:Builder (process-sort-report) in one step. This is accomplished by having the report file sorted internally rather than having the report file written to an external file that is passed to a separate sort step and the sorted report file is passed again to the report step. This optimization method eliminates almost all of the I/O related to the report file as opposed to running each of the process, sort, and report steps as separate job steps. SSP requires more main storage (to accommodate the sort program's requirements). In general, this optimization method should always be selected if sufficient main storage is available.

You can select SSP by specifying SORT INTERNAL on the CONTROL command in the Run Control command group. However, because this is the default, it is not necessary. Conversely, if an external sort and three-step processing is desired, SORT EXTERNAL is specified on the CONTROL command.

**Note:** Each method has specific JCL requirements. See the *VISION:Builder for OS/390 Environment Guide* or *VISION:Builder for VSE Environment Guide* (for VSE users) for more information.

# Report Page Layout

The actual size of a physical page is defined by the height of the paper (which is controlled by the channel 1 printer carriage control) and by the width of the printer. Both of these parameters are outside the control of VISION:Builder and are functions of the particular installation. VISION:Builder generates a logical page (see Figure 9-25).

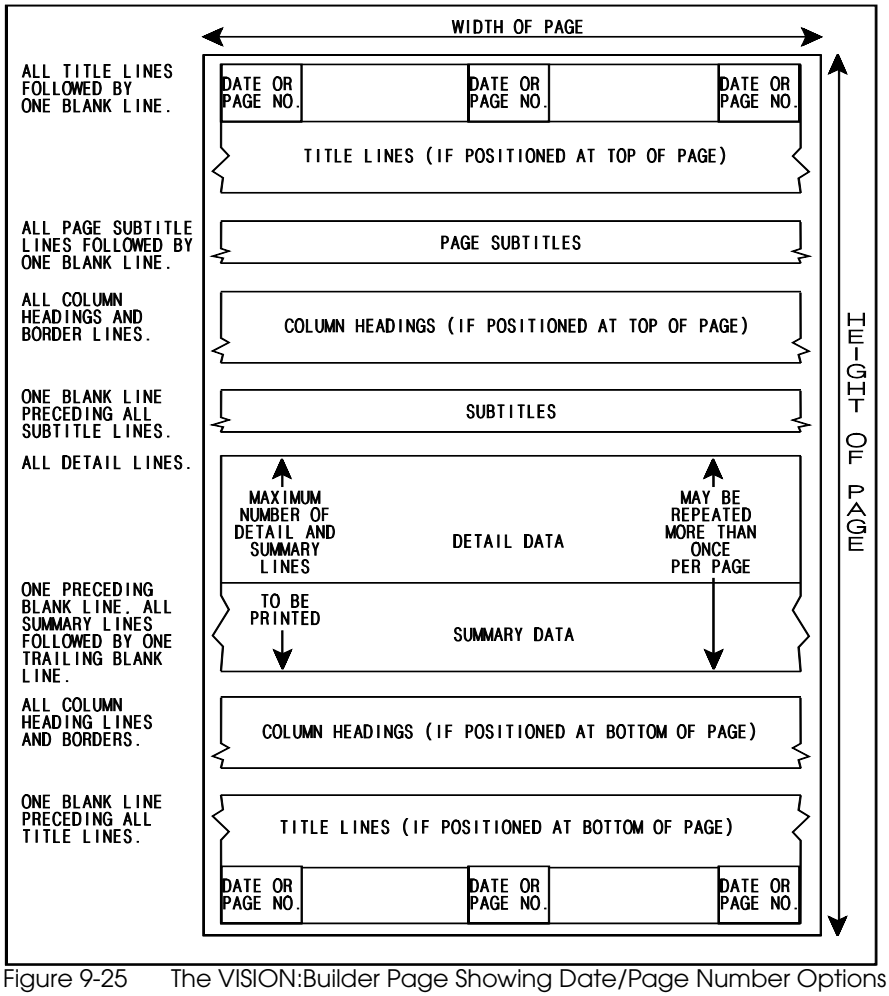



Figure 9-25 The VISION:Builder Page Showing Date/Page Number Options and Relative Placement of Headings

For correct operation, the logical VISION:Builder page must be synchronized with the physical paper size. For most installations, this means that the correct printer carriage control must be loaded.

| | |
|---|---|
| Width of page | The number of columns of data available on a logical page. It does not include the carriage control character. The maximum the user can specify is defined as the M4LIST width in M4PARAMS. By default, this is formatted at the Default Width of Page value. |
| Height of page | The number of physical lines on the page. If VISION:Builder folds a print line, it generates more than one physical line. |

The number of lines printed per page is determined as follows:

- The default number of lines printed is:
  - Page height minus one inch when page height is greater than one inch.
  - Page height minus one line when page height is one inch or less
- The LINESPERPAGE keyword (maximum number of lines per page) on the FORMAT command in the Report command report can be used to override the default number of lines printed; when specified, the entry must be less than the page height. This entry alters the default number of lines printed.

Actual vertical margins on the physical page are determined by the placement of the channel 1 carriage control, relative to the top of the physical page. The maximum number of times a line can fold is controlled by the page height. For example, with a standard size printer page, a setting of channel 1 approximately ½ inch from the top of the physical page results in ½ inch margins at the top and bottom of each physical page; a setting of channel 1 at the top of the page results in a 1 inch margin at the bottom of the page and none at the top of the page.

See Figure 9-25 on page 9-20 when considering that the VISION:Builder logical page is generated as follows:

- All top lines (title and/or column headings) are output.
- The numbers of top lines and bottom lines are subtracted from the LINESPERPAGE keyword.
- The LINESPERPAGE keyword is tested for overflow and remaining detail lines are output.
- Bottom lines are output.
- A Skip to Channel 1 is issued.
- Page width is M4LIST width. You can specify any value less than this value.

Note that the VISION:Builder logical page always contains all requested title lines, all requested column headings lines, and at least one detail line. VISION:Builder uses the following ASA carriage control characters:

| | |
|---|---|
| Blank | - single-space |
| 0 | - double-space |
| - | - triple-space |
| 1 | - skip to channel 1 |

Multiple images on a single page are specified on the FORMAT command with the IMAGES keyword. Number of repeated images = N does not imply that the contents of a logical page are repeated N times but that there are N logical pages horizontally on the physical page. For example, the following values will result in the output shown in Figure 9-26.

| | |
|---|---|
| Height of page | - 16 |
| Width of page | - 10 |
| Number of repeated images | - 4 |

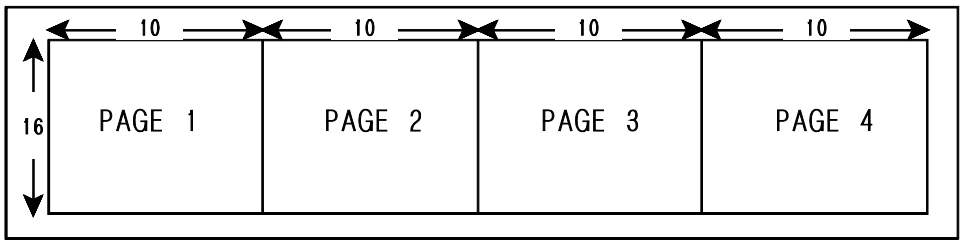



Figure 9-26 Number of Repeated Images

The physical page width must be at least the product of the width and the number of repeated images. When all images have been printed, a skip to channel 1 is issued.

Chapter

# 10 Transactions

**Note:** This chapter only applies to the VISION Builder 4000 model series.

Transaction processing provides techniques to maintain application files. Procedures are fine for minor, general, or infrequent updating; however, as the application grows, minor updating can be inefficient, general updating can be clumsy, and infrequent ad-hoc updating can be prone to error.

Transaction processing, using transaction definitions cataloged in the common library against a file definition, automatically and reliably updates the file. To update, indicate the master file requiring maintenance, supply the file definition of the transaction records, and specify the transaction definitions relating to them.

This chapter discusses the transaction definitions and automatic components of the transaction cycle. Chapter 15, *Advanced Transaction Processing* discusses the customizable transaction cycle components and the advanced transaction processing topics. Chapter 2, *Application Development System* discusses the basic processing cycle and the associated transaction cycle.

Figure 10-1 (repeated from Chapter 2, *Application Development System*) presents a high level view of the transaction cycle. Figure 10-2 shows the transaction processing step. A transaction processing step must contain a transaction file composed of transaction records for updating the old master file. The shaded regions include the transaction file, transaction definitions, transaction work area, and the reject file.

The basic processing flowchart, enhanced in Figure 10-3 on page 10-4, details the processing of the transaction file, transaction definitions, and the reject file. Figure 10-2 and Figure 10-3 together describe the basics of the VISION:Builder automatic transaction process. The flow depicted in the unshaded areas of Figure 10-3 is described in Chapter 5, *The Application Cycle*.

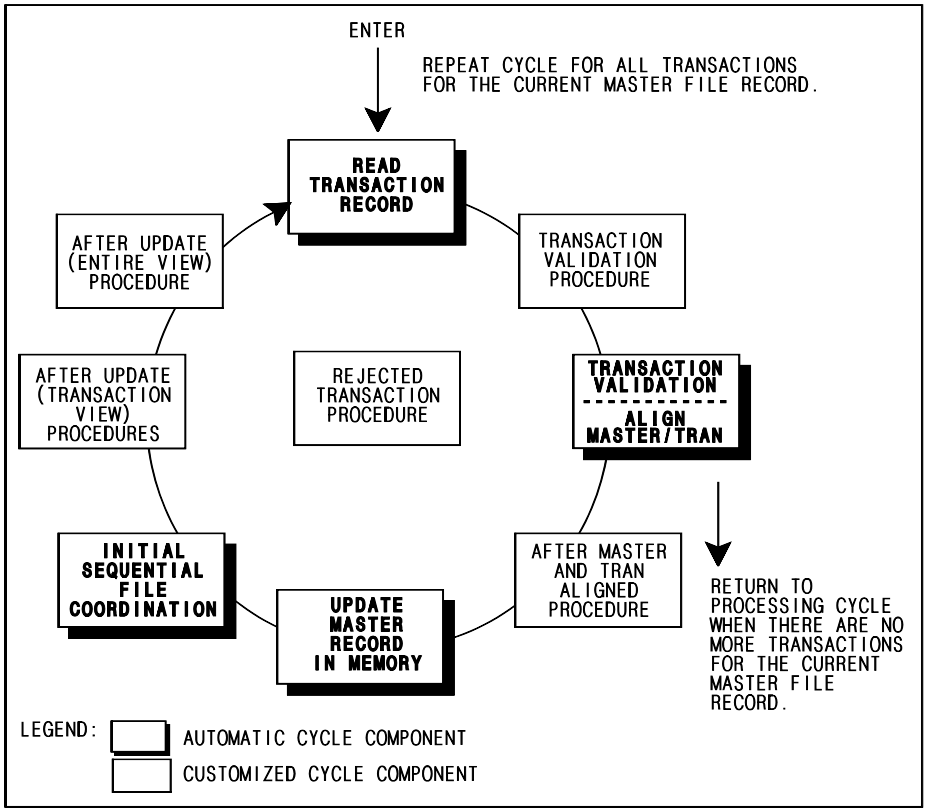



Figure 10-1 Transaction Cycle

**Note:** In Figure 10-2, old and new master file are same physical file if updating-in-place.

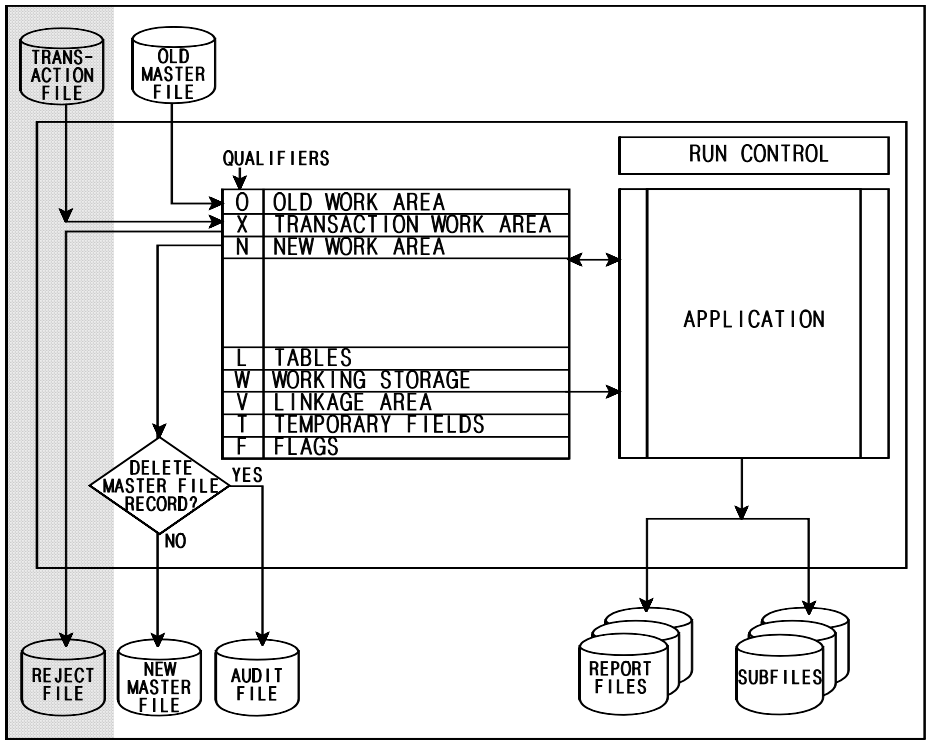



Figure 10-2 Transaction Processing Step

**Note:** In Figure 10-3, TF is transaction record key field value. MF is master record key field value.

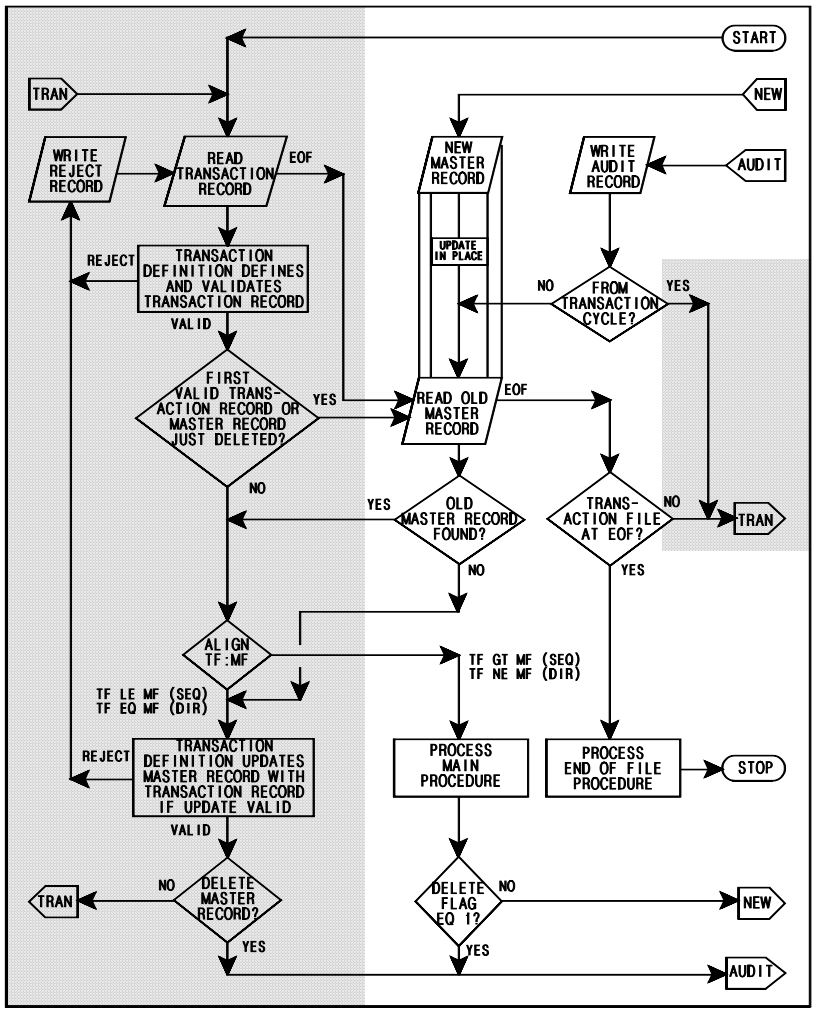



Figure 10-3 Transaction Processing Flow

# Transaction File and Master File Record Flows

In transaction processing, the transaction file is always read sequentially. The master file can be read sequentially or directly (the default is sequential). When the master file is read sequentially, the master file is the driving file and the transaction file must be in ascending sequence by matching master file keys.

When the master file is read directly (randomly), the transaction file becomes the driving file. In this case, only master file records whose keys match the transaction file keys are read. The order of the transaction file is not important; although, the grouping together of transactions for the same master file key will yield the optimum access pattern for the master file.

The following sections describe in more detail the flow for both sequential and direct master file processing flows.

## Transaction File and Sequential Master File Record Flows

VISION:Builder reads a transaction record into the work areas (see Figure 10-3). The transaction record is identified by a transaction definition that validates the transaction record as described in Transaction Definitions on page 10-7. An invalid transaction record is rejected and optionally written to the reject file. Another transaction record is read. If this is the first valid transaction record, the first old master record is read into the work areas.

The sequentially read old master record is always found, unless the old master file is at end of file. The transaction definition checks the alignment of the identified and validated transaction record with the master record. If the transaction record key field value is not greater than the master record key field value and the update is valid, VISION:Builder updates the master record according to the specifications in the transaction definition. If the update is invalid (for example, it attempts to delete a non-existing occurrence or create an existing record or occurrence, and so on), the transaction record is rejected and optionally written to the reject file. Another transaction record is read.

A valid update can delete the master record. In this case, an audit record can be written. Because the transaction cycle is still active (that is, the transaction record key field value is not greater than the master record key field value), another transaction record is read. If the master record is not deleted, VISION:Builder reads another transaction record that is identified by a transaction definition. If the previous transaction record deleted the master record, another old master record is read.

When the value of the transaction record key field is greater than the value of the master record key field, the transaction cycle is exited and the Main (Normal, Type N) procedure or request controls the work areas to report the master record, and so on. As master records are read, they are aligned with the current transaction record. When the value of the master record key field is not less than the value of the transaction record key field, the transaction cycle is reentered just below the align decision diamond.

Transaction processing terminates when the transaction and master file both reach end of file. If the transaction file reaches end of file first, the flow branches to read the old master record. At all subsequent alignments, however, the value of the transaction record key field is assumed as EOF (end of file), which is understood to have the highest key field value.

As a result, the transaction cycle is avoided by the EOF branch after attempting to read a transaction record. Because the align decision diamond always encounters a high value for the transaction record key, the flow branches out of the transaction cycle. The master records with key field values greater than all transaction records are processed as described in Chapter 5, *The Application Cycle* and are available for reporting to the Main (type N) procedures. The run terminates when the master file reaches end of file and the End of File (EOF, Type E) procedures control the work areas.

If the master file reaches end of file first, transaction records can still create master records. As a result, the shaded area on the right side of Figure 10-3 branches to read the transaction file if it is not at end of file. Flow remains in the transaction cycle, except to process created master records with the Main (Normal, Type N) procedures and unless a master record is deleted. In this case, the master file is still at end of file and the transaction file is read again. The run terminates when the transaction file reaches end of file and the End of File (EOF, Type E) procedures control the work areas one last time.

Sequentially updating the master file with the transaction file is very efficient if many master records must be updated; however, it is inefficient if only a few master records require maintenance, because the entire master file is usually read. Also, the transaction and master files must be sorted according to the record key field values that are used to align them. This ensures that the transaction file is not out of sequence and the run is not prematurely terminated.

## Transaction File and Direct Master File Record Flows

VISION:Builder reads a transaction record that is identified and validated by a transaction definition. See Transaction Definitions on page 10-7 for information about transaction definitions. Because this is the first transaction record, the old master file is read directly using the value of the transaction record key field determined by the transaction definition (a record match, create, or delete field).

If the old master record is not found, the transaction record must be identified by a transaction definition that specifies a master record creation. If the master record is found, alignment defaults to TF equals MF (the value of the transaction record key field equals the value of the master record key field). In either case, after the transaction record updates the master record or is rejected, flow remains in the transaction cycle.

After a subsequent transaction record is read, the old master file is not read unless the previous master record was deleted. If it was deleted, flow is described in the previous paragraph. At this point, alignment has a choice between TF equals MF or TF does not equal MF. If TF equals MF, flow remains in the transaction cycle; otherwise, flow proceeds to the Main (Normal, Type N) procedures as in sequential updating. See Transaction File and Sequential Master File Record Flows on page 10-5.

When the master file is directly updated, only the transaction file reaches end of file. At this time (as shown in Figure 10-3), the directly read master file is also assumed to be at end of file. The End of File (EOF, Type E) procedures control the work areas and processing terminates.

Often, it is very useful in large applications, especially those that share files (databases), to update-in-place the master file. In this case, use the UPDATE keyword on the FILE MASTER Command. The associated flow is depicted in Figure 10-3 with the broken lines that join the new master file and the old master file. Old master and new master work areas are used with both sequential updating and update-in-place.

When the master file is directly updated, not every master record passes through the work areas. In fact, only those master records that are updated by the transaction file pass through the work areas. If your application requires all master records (not including deleted master records), update each one or sequentially update the master file. To minimize the number of master file records read and written, sort the transaction file prior to the run according to its record match, create, or delete fields. Thus, each master record is read and written only once.

## Transaction Processing Applications

With VISION:Builder updating facilities, application programs easily and reliably maintain files so that your reports provide up-to-date information. Unlike most procedural application programs that usually conceal the critical updating routines, VISION:Builder applications usually update files using simple and direct transaction definitions that are cataloged in the common library. The transaction definitions can even be grouped so that one file can be maintained by any of several updating functions. Once you understand where the transaction definitions fit in the VISION:Builder automatic transaction processing cycle and you catalog the transaction definitions to relate transaction records to a master file, you can maintain the file very easily.

## Transaction Definitions

Before you can maintain a file using transaction processing, you need the following items cataloged in the common library.

- The master file definition defining the records in the file to be updated. The complete File Definition and Glossary Listing for the Customer Master File (CUSTMAST) used in the following examples is in CUSTMAST File Definition on page D-137.
- The transaction definition describing the updating operations to be performed on the master file. The complete Transaction Definition and Glossary Listing for the Order Entry Transaction Group (ORDENTRY) used in the following examples is in ORDENTRY Transaction Group Definition on page D-148.
- Optionally, a transaction file definition defining the records to be applied to the master file. The complete File Definition and Glossary Listing for the Transaction File (TRANWORK) used in the following examples is in TRANWORK File Definition on page D-145.

The following order-entry application demonstrates transaction definitions that maintain the customer file.

The order-entry transactions only affect the fields shown in the ORDER and ITEMORD segments. The customer file structure is shown in Figure 10-4. The CUSTNO field in the root segment is shown because it aligns the transactions with

a specific master file record. The other root segment fields and the fields in the SHIPINV, ITEMSHIP, and INSTALL segments are not affected by the order-entry transactions.

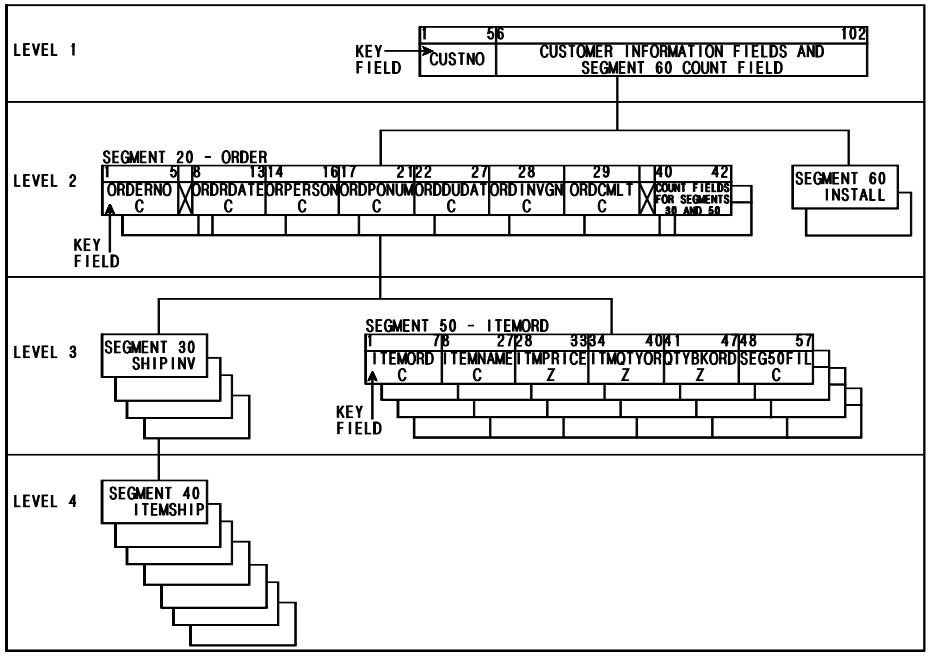



Figure 10-4 The Customer File Structure

The formats of the order-entry transaction records are shown in Figure 10-5. Order-entry requires four transaction record formats: one to add orders, a second to add items to orders, a third to delete orders, and a fourth to delete items from an order.

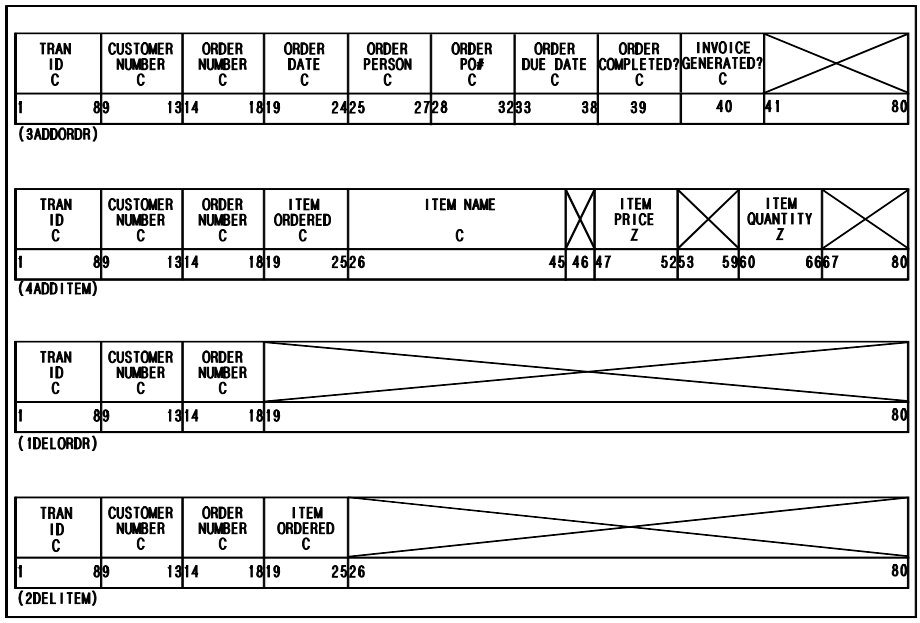



Figure 10-5 Order-Entry Transaction Record Formats

Figure 10-6 shows a glossary listing of some of the fields in the transaction file definition. A file definition for the transaction file is only needed when the transaction fields will be referenced in an application using transaction procedures. For more information on transaction procedures, see Chapter 15, *Advanced Transaction Processing*. For more information about defining files, see Chapter 3, *Defining Data*.

Note that the transaction file is keyed (sequenced) by CUSTNO (customer number) and TRANID (transaction identifier). The transaction identifiers in positions 1-8 of the order-entry transaction records are set up so the delete actions occur before the add actions.

```
                                    DETAILED GLOSSARY BY NAME FOR                                    GLOSSARY PAGE     1
                                        FILE DEFINITION - TRANWORK
DATE CATALOGED - 02/11/02 14:27:06                                          DATE LAST UPDATED - 02/11/02 14:27:06
CATALOGED BY -                                                              LAST UPDATED BY -
EXPIRATION DATE -                                                           DATE LAST USED -
FILE IDENTIFICATION =                                                       RECORD FORMAT = FIXED BLOCKED
NUMBER OF SEGMENTS IN FILE = 1                                              RECORD SIZE = 80
NUMBER OF FIELDS IN FILE = 20                                               BLOCK SIZE = 800

                                      ******************************
                                      *    SEGMENT   1, LEVEL 1    *
                                      *   SEGMENT NAME = WORK01    *
                                      ******************************
SEGMENT OCCURS N TIMES = 1                                  KEY FIELD 1 = CUSTNO     TYPE = C   LENGTH = 5
SEGMENT SIZE = 80
NUMBER OF FIELDS IN SEGMENT = 20
-------------------------------------------------------------------------------------------------------------------------
FIELD   FIELD   FIELD    FIELD   FIELD    DEC    CNT FIELD  EDIT CODES    EDIT    OUTPUT   LINE  DATA BASE NAME/
NAME    TYPE   LOCATION  LENGTH  RNDING  PLACES  FOR SGMT   ( ) ( ) ( )  LENGTH   WIDTH     NO   *** COLUMN HEADING ***
-------------------------------------------------------------------------------------------------------------------------
TRANID    C        1        8                                              8        8
                                                                                          1    ***     TRANS      ***
                                                                                          2    ***      ID        ***
CUSTNO    C        9        5                                              5        5
                                                                                          1    ***     CUST       ***
                                                                                          2    ***     NO.        ***
ITM5ORD1  C       19        7                                              7        7
```

Figure 10-6 Transaction File Definition Glossary Listing (Page 1 of 2)

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | | | 1 | *** | ITEM | *** |
| | | | | | | 2 | *** | ORDR 1 | *** |
| ITM5QTY1 | Z | 26 | 5 | 7 | 7 | | | | |
| | | | | | | 1 | *** | ITEM | *** |
| | | | | | | 2 | *** | QTY 1 | *** |
| ORDERNO | C | 14 | 5 | 5 | 5 | | | | |
| | | | | | | 1 | *** | ORDER | *** |
| | | | | | | 2 | *** | NO. | *** |
| ORDRCOMP | C | 40 | 1 | 1 | 3 | | | | |
| | | | | | | 1 | *** | INV | *** |
| | | | | | | 2 | *** | CMP | *** |
| ORDRDATE | C | 19 | 6 | 6 | 6 | | | | |
| | | | | | | 1 | *** | ORDER | *** |
| | | | | | | 2 | *** | DATE | *** |
| ORDRDUED | C | 33 | 6 | 6 | 6 | | | | |
| | | | | | | 1 | *** | DUE | *** |
| | | | | | | 2 | *** | DATE | *** |
| ORDRINVG | C | 39 | 1 | 1 | 3 | | | | |
| | | | | | | 1 | *** | INV | *** |
| | | | | | | 2 | *** | GEN | *** |
| ORDRPERS | C | 25 | 3 | 3 | 6 | | | | |
| | | | | | | 1 | *** | ORDER | *** |
| | | | | | | 2 | *** | PERSON | *** |
| ORDRPONO | C | 28 | 5 | 5 | 5 | | | | |
| | | | | | | 1 | *** | ORDER | *** |
| | | | | | | 2 | *** | PO | *** |

Figure 10-6 Transaction File Definition Glossary Listing (Page 2 of 2)

The complete File Definition and Glossary Listing for the Transaction File TRANWORK is in TRANWORK File Definition on page D-145.

The file definitions for both the customer and transaction files define the records, fields, and segments in these files. The file definitions do not specify how to apply the transaction records. For this, you need a transaction definition that relates the customer and transaction files by field name. The transaction definition against the customer file identifies the actions of the transaction file against the master file and validates the transaction data.

Each transaction definition must belong to a transaction group. Transaction groups describe the updating operations for the master file and can include several transaction definitions. Transaction groups specify the following:

- Master file to be updated.
- Identification of the transaction file record types.
- Actions to be taken against master file records.
- Transaction data input validation criteria.

Each transaction specifies one or two codes (transaction identifiers) that are used to recognize data in the transaction file. The data is applied to the master file according to the actions specified in the transaction definition.

Identification of the transaction includes the following:

- The name of the transaction file field containing the transaction code.
- An equal or not equal test for the transaction code to determine whether the transaction is to be applied.
- A transaction code value to be used to identify the transaction data.

In the ORDENTRY Transaction Group there are five transaction record types that are identified as 3ADDORDR, 4ADDITEM, 5ADDITEM, 1DELORDR, and 2DELITEM. Figure 10-7 shows a portion of the transaction group glossary where the transaction identifiers are displayed.

```
                                    GLOSSARY BY LEVEL NUMBER FOR                                    GLOSSARY PAGE     1
                                     TRANSACTION GROUP - ORDENTRY

DATE CATALOGED - 02/11/02 14:27:06                                  DATE LAST UPDATED - 02/11/02 14:27:06
CATALOGED BY -                                                      LAST UPDATED BY -
EXPIRATION DATE -                                                   DATE LAST USED -
MASTER FILE NAME = CUSTMAST                                         RECORD FORMAT = FIXED BLOCKED
NUMBER OF IDENTIFIERS IN GROUP = 5                                  RECORD SIZE = 80
NUMBER OF ACTIONS IN GROUP = 23                                     BLOCK SIZE = 800

                                  ************************************
                                  *  1  OF  5  IDENTIFIERS IN GROUP  *
                                  ************************************
**********    ACTIONS IN IDENTIFIER = 8          IDENTIFIER ONE = (3ADD)          IDENTIFIER TWO = (ORDR)
*3ADDORDR*    DEFAULT CREATE = NO                IDENTIFIER LOCATION = 1          IDENTIFIER LOCATION = 5
**********    DEFAULT INSERT = NO                IDENTIFIER LENGTH = 4            IDENTIFIER LENGTH = 4
                                                 IDENTIFIED IF EQUAL              IDENTIFIED IF EQUAL
--------------------------------------------------------------------------------------------------------------------------------
                                  ************************************
                                  *  2  OF  5  IDENTIFIERS IN GROUP  *
                                  ************************************
**********    ACTIONS IN IDENTIFIER = 6          IDENTIFIER ONE = (4ADD)          IDENTIFIER TWO = (ITEM)
*4ADDITEM*    DEFAULT CREATE = NO                IDENTIFIER LOCATION = 1          IDENTIFIER LOCATION = 5
**********    DEFAULT INSERT AT LEVEL 3          IDENTIFIER LENGTH = 4            IDENTIFIER LENGTH = 4
                                                 IDENTIFIED IF EQUAL              IDENTIFIED IF EQUAL
--------------------------------------------------------------------------------------------------------------------------------
                                  ************************************
                                  *  3  OF  5  IDENTIFIERS IN GROUP  *
                                  ************************************
**********    ACTIONS IN IDENTIFIER = 4          IDENTIFIER ONE = (5ADD)          IDENTIFIER TWO = (ITEM)
*5ADDITEM*    DEFAULT CREATE = NO                IDENTIFIER LOCATION = 1          IDENTIFIER LOCATION = 5
**********    DEFAULT INSERT AT LEVEL 3          IDENTIFIER LENGTH = 4            IDENTIFIER LENGTH = 4
                                                 IDENTIFIED IF EQUAL              IDENTIFIED IF EQUAL
--------------------------------------------------------------------------------------------------------------------------------
                                  ************************************
                                  *  4  OF  5  IDENTIFIERS IN GROUP  *
                                  ************************************
**********    ACTIONS IN IDENTIFIER = 2          IDENTIFIER ONE = (1DEL)          IDENTIFIER TWO = (ORDR)
*1DELORDR*    DEFAULT CREATE = NO                IDENTIFIER LOCATION = 1          IDENTIFIER LOCATION = 5
**********    DEFAULT INSERT = NO                IDENTIFIER LENGTH = 4            IDENTIFIER LENGTH = 4
                                                 IDENTIFIED IF EQUAL              IDENTIFIED IF EQUAL
--------------------------------------------------------------------------------------------------------------------------------
                                  ************************************
                                  *  5  OF  5  IDENTIFIERS IN GROUP  *
                                  ************************************
**********    ACTIONS IN IDENTIFIER = 3          IDENTIFIER ONE = (2DEL)          IDENTIFIER TWO = (ITEM)
*2DELITEM*    DEFAULT CREATE = NO                IDENTIFIER LOCATION = 1          IDENTIFIER LOCATION = 5
**********    DEFAULT INSERT = NO                IDENTIFIER LENGTH = 4            IDENTIFIER LENGTH = 4
                                                 IDENTIFIED IF EQUAL              IDENTIFIED IF EQUAL
--------------------------------------------------------------------------------------------------------------------------------
```

Figure 10-7 Transaction Identifiers for ORDENTRY Transaction Group

In the glossary listing there are two transaction identifier areas for each transaction type. The entries in these areas are used to identify transaction records. The information in identifier one and two is combined to form the values used to identify transaction records that add orders and items to the CUSTMAST master file.

Each identifier consists of four entries: an identifier code value, the starting location of the code within the transaction record, the length of the code, and operation to use when comparing for the code. Refer to the *VISION:Builder Reference Summary* for complete details for coding the TD (Transaction Definition) statement.

The complete Transaction Definition and Glossary Listing for the Order Entry Transaction Group (ORDENTRY) is in ORDENTRY Transaction Group Definition on page D-148.

In our ORDENTRY transaction group, positions 1 to 8 of the transaction record must contain the identifiers 3ADDORDR, 4ADDITEM, 5ADDITEM, 1DELORDR, and 2DELITEM.

Once the transaction file record types are identified, the actions for each identifier can be defined. These actions will specify how the transaction data is processed and applied to the fields in the master file.

The Definition of the actions in a transaction can include the following:

- The location of the transaction data identified for updating.
- The segment and field actions for the master file field.
- The name of the master file field identified for updating by the transaction data.
- Specifications for validating transaction data.

The transaction definition for the 3ADDORDR transaction record type is shown in Figure 10-8.

```
                                    ************************************
                                    *  1 OF 5 IDENTIFIERS IN GROUP   *
                                    ************************************
**********   ACTIONS IN IDENTIFIER = 8          IDENTIFIER ONE = (3ADD)          IDENTIFIER TWO = (ORDR)
*3ADDORDR*   DEFAULT CREATE = NO                IDENTIFIER LOCATION = 1          IDENTIFIER LOCATION = 5
**********   DEFAULT INSERT = NO                IDENTIFIER LENGTH = 4            IDENTIFIER LENGTH = 4
                                                IDENTIFIED IF EQUAL              IDENTIFIED IF EQUAL
-------------------------------------------------------------------------------------------------------------------------
MASTER  ..........TRANSACTION DATA..........
FILE    FIELD     FIELD     FIELD    DECIMAL   ACTION   LEVEL          MINIMUM                  MAXIMUM
FIELD   TYPE     LOCATION   LENGTH   PLACES     CODE    NUMBER          VALUE                    VALUE
-------------------------------------------------------------------------------------------------------------------------
CUSTNO    C          9        5                 M        1          EDIT = Y9999
ORDERNO   C         14        5                 I        2
ORDRDATE  C         19        6                 R        2                    DATE VALIDATION
ORPERSON  C         25        3                 R        2          EDIT = -9-9-9
ORDPONUM  C         28        5                 R        2
ORDDUDAT  C         33        6                 R        2                    DATE VALIDATION
ORDINVQN  C         39        1                 R        2          EDIT = Z
ORDCMPLT  C         40        1                 R        2
```

Figure 10-8 3ADDORDR Transaction Definition Glossary Listing

The glossary listing shows the transaction data location and type along with the actions to be applied against the named master file fields. Any specific validation to be performed on the transaction data is also shown in the glossary.

The action against CUSTNO is M (MATCH). This means the transaction file data must match the master file field which aligns the transaction and master records. The CUSTNO field is also validated, as shown by EDIT = Y9999; the first position can be alpha or numeric and the remaining positions must be numeric. Allowable validation entries include date, pattern, and min/max.

If a match on the CUSTNO field occurs, the remaining actions are performed.

- The ORDERNO field is processed next. If a match between the transaction and master file does not occur for this field, the I (INSERT) action code indicates that the segment is to be inserted into the master file. If a match occurs, the transaction is rejected as it would create a duplicate ORDER segment. Note that conditional inserts are also possible (see discussion of actions at the end of this section).

- The R (REPLACE) action against the remaining fields replaces the master file fields with the transaction fields if they meet any validation criteria and the transaction fields are not empty.
- The ORPERSON field contains validation criteria that must be met before it can be replaced. ORPERSON is validated by a pattern of EDIT = -9-9-9. This ensures that the value in the transaction field does not contain numeric information. If it does, the transaction record is rejected.
- The ORDINVGN field also contains validation criteria, EDIT = Z. This field must contain an alphabetic character or it is rejected.

In this way, the 3ADDORDR transaction definition primarily updates the ORDER segment in the customer file.

**Note:** In transaction processing, if a master record is created or a segment inserted, all fields except the record or segment key fields are initialized to blanks for character string data and zeros for numeric data. When fields overlap, they are initialized based on the data type of the last overlapping field defined.

Remember that the R (REPLACE) action code is applicable only if the transaction record data is not empty. This facilitates transaction records that update only part of the master fields referenced by the transaction definition while preserving the master fields corresponding to empty transaction record fields. The 3ADDORDR transaction definition is limited to the record path, beginning with the root segment of the matched master record and ending with the inserted ORDER segment occurrence.

The 4ADDITEM, 1DELORDR, and 2DELITEM transaction definitions are shown in Figure 10-9 through Figure 10-11. These transaction definitions attempt to identify the transaction record in the ORDENTRY group. Because they are in the ORDENTRY group (see Figure 10-7), they identify and apply transaction records with the same characteristics as those defined by the ADDORDR transaction definition.

Note that the transaction record type 5ADDITEM is discussed in Chapter 15, *Advanced Transaction Processing*.

Each transaction definition identifies transaction records with compound identifiers like the ADDORDR transaction definition. Once the transaction record is identified with a format like the one in Figure 10-5, the transaction record fields are validated according to the criteria specified in the transaction definition.

- In all transaction records identified by the 4ADDITEM, 1DELORDR, or 2DELITEM transaction definitions, the field CUSTNO must begin with an alpha or numeric character and the last four positions must be numeric.
- The ITMPRICE field must be numeric with a value between 0 and 300.

Any transaction record not meeting these criteria is automatically rejected.

The identifying transaction definition converts the transaction record data to the data types of the master file fields specified. Once each transaction record data is verified and converted, it is aligned with the new work area and applied accordingly.

The transaction definition for the 4ADDITEM transaction record type is shown in Figure 10-9.

```
                                    GLOSSARY BY LEVEL NUMBER FOR                                    GLOSSARY PAGE      2
                                    TRANSACTION GROUP - ORDENTRY

                                  ************************************
                                  *  2 OF 5 IDENTIFIERS IN GROUP     *
                                  ************************************
**********    ACTIONS IN IDENTIFIER = 6          IDENTIFIER ONE = (4ADD)          IDENTIFIER TWO = (ITEM)
*4ADDITEM*    DEFAULT CREATE = NO                IDENTIFIER LOCATION = 1          IDENTIFIER LOCATION = 5
**********    DEFAULT INSERT AT LEVEL 3          IDENTIFIER LENGTH = 4            IDENTIFIER LENGTH = 4
                                                 IDENTIFIED IF EQUAL              IDENTIFIED IF EQUAL
-------------------------------------------------------------------------------------------------------------------------
MASTER  .........TRANSACTION DATA.........
FILE    FIELD     FIELD     FIELD    DECIMAL   ACTION   LEVEL          MINIMUM                 MAXIMUM
FIELD   TYPE     LOCATION   LENGTH   PLACES    CODE     NUMBER         VALUE                   VALUE
-------------------------------------------------------------------------------------------------------------------------
CUSTNO    C          9        5                M        1          EDIT = Y9999
ORDERNO   C         14        5                M        2
ITEMORD   C         19        7                M        3
ITEMNAME  C         26       20                R        3
ITMPRICE  Z         47        6       2        R        3      ***0.00          ***     ***300.0         ***
ITMQTYOR  Z         60        7                R        3
```

Figure 10-9 ADDITEM Transaction Definition Glossary Listing

The 4ADDITEM transaction definition applies transaction record fields to the customer file according to the action codes described below.

- Transaction data is matched at the root level to the master file CUSTNO field.
- Transaction data is matched at level 2 to the master file ORDERNO field.
- Transaction data is updated at level 3 to the ITEMORD field if a match occurs. It is inserted if it does not exist.
- Transaction data, if not empty, is placed in the ITEMNAME field of the matched ITEMORD segment occurrence.
- Transaction data, if not empty, is placed in the ITMPRICE field of the matched ITEMORD segment occurrence.
- Transaction data, if not empty, is placed in the ITMQTYOR field of the matched ITEMORD segment occurrence.

The following describes the actions specified in the transaction definition.

- Line 1 aligns the customer and transaction file records on CUSTNO.
- Line 2 matches the transaction file data to an ORDERNO segment occurrence. If no match is found, the transaction is rejected because it is at level 2 not level 3.

- Once an ORDER segment is matched, Line 3 matches the transaction file data to the ITEMORD segment occurrence; however, if no match is found, a level 3 occurrence is inserted.
- Lines 4, 5, and 6 replace the specified transaction record positions, if not blank, into the ITEMNAME, ITMPRICE, and ITMQTYOR fields of the matched or created ITEMORD segment occurrence.

The 4ADDITEM transaction definition is limited to the matched customer record's ORDER segment occurrence and to the matched or created ITEMORD segment occurrence. This is a single-path in the customer record.

The transaction definition for the 1DELORDR transaction record type is shown in Figure 10-10.

```
                                    GLOSSARY BY LEVEL NUMBER FOR                                    GLOSSARY PAGE      4
                                    TRANSACTION GROUP - ORDENTRY

                                  ************************************
                                  *  4 OF 5 IDENTIFIERS IN GROUP   *
                                  ************************************
**********   ACTIONS IN IDENTIFIER = 2          IDENTIFIER ONE = (1DEL)          IDENTIFIER TWO = (ORDR)
*1DELORDR*   DEFAULT CREATE = NO                IDENTIFIER LOCATION = 1          IDENTIFIER LOCATION = 5
**********   DEFAULT INSERT = NO                IDENTIFIER LENGTH = 4            IDENTIFIER LENGTH = 4
                                                IDENTIFIED IF EQUAL              IDENTIFIED IF EQUAL
-------------------------------------------------------------------------------------------------------------------------
MASTER  ..........TRANSACTION DATA..........
FILE    FIELD     FIELD      FIELD    DECIMAL   ACTION   LEVEL          MINIMUM                MAXIMUM
FIELD   TYPE     LOCATION    LENGTH   PLACES     CODE    NUMBER          VALUE                  VALUE
-------------------------------------------------------------------------------------------------------------------------
CUSTNO    C          9         5                M        1          EDIT = Y9999
ORDERNO   C         14         5                E        2
```

Figure 10-10 1DELORDR Transaction Definition Glossary Listing

The 1DELORDR transaction definition applies the following transaction record data fields to the customer file using the specified action codes.

- Transaction data is matched at level 1 to the master file CUSTNO field. This aligns the customer and transaction file records.
- Transaction data is matched at level 2 to the master file ORDERNO segment occurrence. If matched, the occurrence is deleted. If no match is found, the transaction record is rejected.

The 1DELORDR transaction definition is limited to the matched customer record's ORDER segment occurrence. This is a single-path in the customer record.

The transaction definition for the 2DELITEM transaction record type is shown in Figure 10-11.

```
                                    GLOSSARY BY LEVEL NUMBER FOR                                    GLOSSARY PAGE      5
                                    TRANSACTION GROUP - ORDENTRY

                                  ************************************
                                  *  5 OF 5 IDENTIFIERS IN GROUP   *
                                  ************************************
**********   ACTIONS IN IDENTIFIER = 3          IDENTIFIER ONE = (2DEL)          IDENTIFIER TWO = (ITEM)
*2DELITEM*   DEFAULT CREATE = NO                IDENTIFIER LOCATION = 1          IDENTIFIER LOCATION = 5
**********   DEFAULT INSERT = NO                IDENTIFIER LENGTH = 4            IDENTIFIER LENGTH = 4
                                                IDENTIFIED IF EQUAL              IDENTIFIED IF EQUAL
```

Figure 10-11 2DELITEM Transaction Definition Glossary Listing

```
-------------------------------------------------------------------------------------------------------------------
MASTER  ..........TRANSACTION DATA..........
FILE    FIELD     FIELD     FIELD    DECIMAL   ACTION   LEVEL          MINIMUM                 MAXIMUM
FIELD   TYPE    LOCATION    LENGTH   PLACES     CODE    NUMBER          VALUE                   VALUE
-------------------------------------------------------------------------------------------------------------------
CUSTNO    C          9        5                 M        1          EDIT = Y9999
ORDERNO   C         14        5                 M        2
ITEMORD   C         19        7                 E        3
```

Figure 10-11 2DELITEM Transaction Definition Glossary Listing

The 2DELITEM transaction definition is similar to the 1DELORDR transaction definition, except it deletes occurrences at level 3. DELITEM is limited to the matched customer record's ORDERNO and ITEMORD segment occurrences. This is a single-path in the customer record.

A transaction definition can, at most, maintain one path in the master record. That is, a transaction definition cannot apply a transaction record to more than one occurrence of a segment or to two segments at the same record level. See Chapter 15, *Advanced Transaction Processing* for more information.

The following is a explanation of action codes that are available when coding transaction definitions. Refer to the *VISION:Builder Reference Summary* for details on coding the TD and TL Transaction Definition Statements..

| Action | Explanation |
|---|---|
| MATCH (M) | Matches the transaction field to the master field. This will align the transaction record with the master record at level 1 or with the segment occurrence at a lower level, when associated with a key field. |
| CREATE (C) | This is used to create a new master file record level 1 segment if no match is found. If a match is found, the transaction record is rejected. |
| INSERT (I) | Inserts a lower level segment occurrence in the matched master record and uses the transaction field value as its key field. If the occurrence already exists, the transaction record is rejected. This action code requires matches, inserts, or creates at all higher levels. |
| DELETE (D) (E) | Deletes the master record segment occurrence with a key field value equal to the value of the transaction field (subordinate segment occurrences are also deleted). The D action code is used for entire record deletes, level 1 key reference. The E action code is used for level 2 – 9 segment occurrence deletes. If the segment occurrence is not found, the transaction record is rejected. This action code requires matches at all higher levels. |
| ADD TO (A) | Adds the value of the transaction field to the value of the master field and puts the sum in the master field. This action code requires matches at all higher levels. |

| Action | Explanation |
|---|---|
| SUBTRACT FROM (S) | Subtracts the value of the transaction field from the value of the master field and puts the difference in the master field. This action code requires matches at all higher levels. |
| REPLACE (P) | Places the value of the transaction field, empty or not, in the master field. This action code requires matches at all higher levels. |
| REPLACE (R) | Places the value of the transaction field, if not empty, in the master field. This action code requires matches at all higher levels and is convenient for transaction definitions that update master fields selected by the transaction record. |
| CLEAR (B) | Clears the master field if the transaction field is not empty. Character fields are set to blank, numeric fields are set to zero, and variable length fields are set to zero length. If the transaction record field is empty, this action code has no effect on the master file. This action code requires matches at all higher levels. |
| CONCATENATE (X) | Concatenates the value of the transaction field to the value of a variable length (type V) master field. It retains leading blanks in the transaction field but does not retain trailing blanks. This action code requires matches at all higher levels and a variable length master field. |

There is an application example that shows how the Customer Master File (CUSTMAST) is created. The application coding is in Application Coding on page D-158.

The application uses the CUSTADDS Transaction Group to create a Customer Record and the ORDENTRY Transaction Group to add the order and item information.

Chapter

# 11 Requests

Requests provide the same functionality as procedure, report, and subfile objects but the procedural part of a request is coded in a fixed format syntax as opposed to the freeform syntax of ASL used in procedures. Refer to the *VISION:Builder Reference Summary* for a full description of the fixed format syntax.

Requests refer to the combined set of specifications for procedural logic, reports, and subfiles. The combined set is assigned a unique name within an application. In other words, a request can contain embedded within it procedural logic, as well as report and subfile specifications. Within VISION:Workbench for DOS and VISION:Workbench for ISPF, reports and subfiles are distinct objects and each must be assigned its own individual name. The term request is used to refer only to the object containing procedural (PR) statements.

This chapter describes the use of requests, as well as the structure of requests and their use in an application. Chapter 5, *The Application Cycle* discusses the use of work areas with regard to procedures, reports, and subfiles. Requests conform to these rules in an identical fashion. In addition, requests can be substituted for procedures within the basic application cycle.

## Request Sections

Request statements are grouped into two sections:

- PR Section — The set of request statements that select or modify work areas, specify branches, or generate calls.
- Rn Section — A set of request statements that reference work area fields (or records) to produce or format a report or a subfile. Each Rn set, where n=1-9, refers to an output selected from this request.

These definitions allow up to one PR section per request and up to nine Rn sections per request. The PR section is used to modify and logically select work area records. The Rn section is used to produce reports or subfiles. Each Rn section can produce one report and one subfile. A PR section and the Rn sections can be part of the same request.

With these definitions, the general structure of a request can be described as shown in Figure 11-1. The figure shows the top of the request preceding its only PR section, followed by the output of the request. A double-headed arrow joins the work areas and the PR section because the PR section can both reference and modify the work areas. The Rn sections follow the output of the request and they are joined to the work areas by a one-headed arrow only because they can only reference the work areas for their report or subfile records.

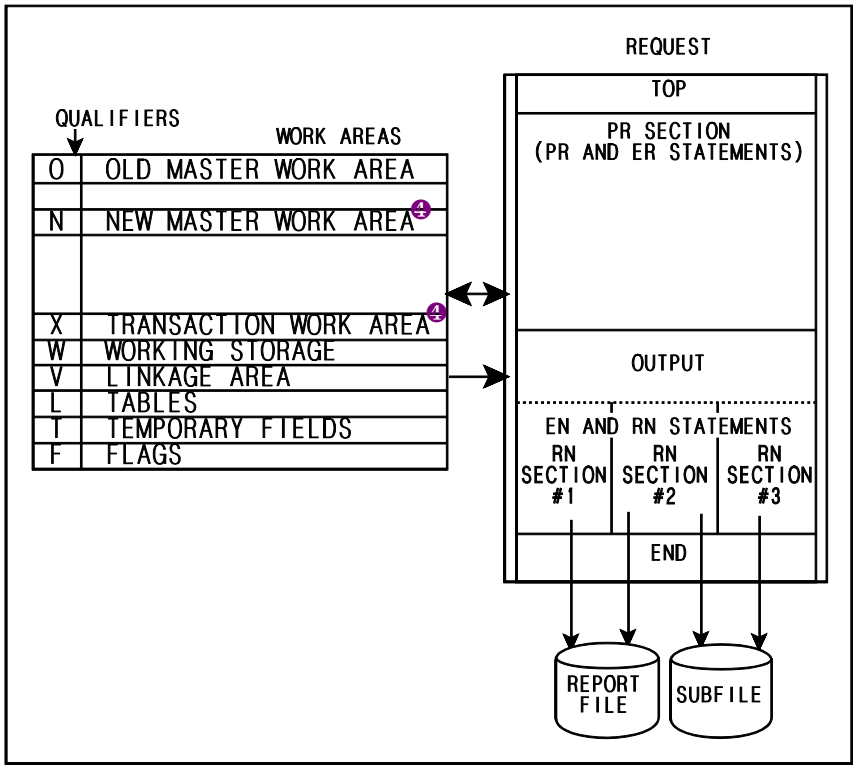



Figure 11-1 Request Sections Showing the Top, Output, and End of the Request

Figure 11-2 shows a request named SAMPLE with one PR section and three Rn sections.

```
                                                  ******************************
                                                  *  REQUEST NAME - SAMPLE     *
                                                  *  INPUT STREAM REQUEST      *
                                                  ******************************
-------------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                                  MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID            ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM       73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------------
(ER)                                                                                  (N)                                          4
-------------------------------------------------------------------------------------------------------------------------------------
STMT SEQ  LOG CON .....OPERAND-A..... OPER  ............OPERAND-B.............  ......RESULT...... PARTIAL FIELD  SEQUENCE   LIST
TYPE NO. LEV CTR QLF FIELD   SEG-LVL  ATN   QLF FIELD   SEG-LVL  (OR CONSTANT)   QLF FIELD  SEG-LVL  STRT NUMB OPR  73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------------
(PR) (010)        (N,NEWFLDA ) 01-1   (EQ)   (C,ACME INC                      )                                                  5
(PR) (030)                            (R )   (C,REPORTED BY SAMPLE REQUEST     ) (N,NEWFLDB ) 01-1                                6
(PR) (060)        (T,COUNTER )        (+ )   (D,1                              ) (T,COUNTER )                                     7
(PR) (070)        (T,COUNTER )        (LE)   (D,5                              )                                                  8
(PR) (080)                            (NS)     (100                            )                                                  9
(PR) (090)                            (GO)     (OUTPUT                         )                                                 10
(PR) (100)                            (R )   (C,E                              ) (F,EOF     )         ( 1) ( 1) (C)               11
```

Figure 11-2 Sample Request Showing Request Structure (Page 1 of 2)

```
(PR) (110)                          (GO)      (END                          )                                                                  12
-------------------------------------------------------------------------------------------------------------------------------------------------
          SP Q                  E N S D C S T C C M M A G D Q PCT/RAT      P                PARTIAL   SEQUENCE   LIST
STMT SEQ  BF L  FIELD   SEG-    N P E E T U 0 U N A I V M E L FIELD   RA- C     PICTURE    STRT NO   73 TO 80   SEQ
TYPE NO.  COL F  NAME     LVL   D R Q S L B T M T X N G R C F NAME    TIO T      EDIT      CHR CHR
-------------------------------------------------------------------------------------------------------------------------------------------------
(R1)        (N,NEWFLDA ) 01-1                                                                                                              13
(R1)        (N,NEWFLDB ) 01-1                                                                                                              14
(R2)        (T,COUNTER )                                                                                                                   15
(R2)        (F,EOF     )                                                                                                                   16
(R3)        (N,NEWFLDA ) 01-1                                                                                                              17
(R3)        (N,NEWFLDB ) 01-1                                                                                                              18
(R3)        (T,COUNTER )                                                                                                                   19
(R3)        (F,EOF     )                                                                                                                   20
```

Figure 11-2 Sample Request Showing Request Structure (Page 2 of 2)

The PR section selects records from the new work area (using the N qualifier or blanks) that have a NEWFLDA value equal to ACME INC. Any new work area records not meeting this requirement direct VISION:Builder to branch around the remaining PR section statements and all the Rn section statements in the request SAMPLE. VISION:Builder assumes a branch to the end of the request after each logical statement for which an explicit NS or GS is not otherwise coded. Request branching is described in Request Branching on page 11-5.

Statement number 80 (SEQNO=80) provides an explicit branch when statement number 70 is false. If the temporary field COUNTER is greater than 5, VISION:Builder branches to statement 100; otherwise, VISION:Builder branches to the output of the request. In all, the PR section references the fields NEWFLDA and COUNTER in the new and temporary work areas using the N and T qualifiers, respectively.

The PR section also modifies the fields NEWFLDB, COUNTER, and EOF in the new, temporary, and flag work areas, respectively. These fields are listed as result fields on the PR statements numbered 30, 60, and 100, respectively.

The flag field EOF is only partially modified by statement 100 because of the partial field entries. Refer to the PR Statement Partial Field section and the Rn Statement Partial Field section of the *VISION:Builder Reference Summary*. This statement forces the old master file to end of file to shorten the application run. For more details concerning PR statement coding, refer to the *VISION:Builder Reference Summary*.

The R1 section references the NEWFLDA and NEWFLDB fields in the new work area using the N qualifier. The values of these fields appear on the report of the R1 section. The R2 section references the COUNTER and EOF fields in the temporary and flag field work areas using the T and F qualifiers, respectively. The values of these fields appear on the report of the R2 section. The R3 section references all four fields in the new, temporary, and flag field work areas that were referenced by the R1 and R2 sections. In all, three reports are produced by the request SAMPLE. These reports, however, are restricted to work area field values that are concurrent with a NEWFLDA value of ACME INC.

# Request Types

In basic master file processing (described in Chapter 5, *The Application Cycle*), VISION:Builder can be customized by the four request types defined below:

| | |
|---|---|
| Type N Requests | Type N requests can reference or modify the work areas while each master record occupies the old/new work areas and before the old master file reaches end of file. |
| Type A Requests | Type A requests can reference or modify the work areas while each master record occupies the old/new work areas and are executed once more after the old master file reaches end of file. When the old master file is at end of file, the old and new work areas are unavailable. Note that type A requests cannot be produced on VISION:Workbench for DOS. |
| Type E Requests | Type E requests can reference or modify the work areas once and are executed after the old master file reaches end of file. |
| Type S Requests | Type S requests can reference or modify the work areas when called by another request. Type S requests are subroutine requests, described in Subroutine Requests and Subroutine Sets on page 11-11. |

With four different request types, you can compose your applications with greater flexibility. Types N and A requests control the work areas at least once for each master record unless they are branched around (this is described in Request Branching on page 11-5). Type S requests provide you with a subroutine facility to reduce the number of redundant statements because they can be called by several requests. Types A and E requests control the work areas after the master file reaches end of file and can produce reports that summarize the application run. Advanced transaction processing is described in Chapter 15, *Advanced Transaction Processing*.

# Request Sequencing

Because requests can modify the work areas, the sequence in which they reference the work areas is important. Requests are sequenced primarily by the type you assign to them (for example type N, A, E, or S) and, secondarily, in the order you enter them. The sequencing can be affected, however, by the branching described in Request Branching on page 11-5.

Because subroutine requests reference work areas only when called by other requests, you can enter them in any order. However, subroutine request sets (multiple subroutine requests with a common set name) are processed in the order you enter them and reference the work areas with only one subroutine request set call.

Subroutine request sets are sequenced apart from request types N and A, which always reference the work areas unless branched around. Subroutine requests return work area control to the request that called them using a GO RETURN, GS RETURN, or NS RETURN procedure statement or automatically after the end of the subroutine request or the last subroutine request set. Also, no subroutine request can call itself, even indirectly (recursion is not allowed). Subroutine requests are also described in Chapter 8, *Data Structure Handling*.

Each type A or E request references the work areas only once after the old master file reaches end of file (EOF flag master partial field equals E). Type E requests cannot reference old or new work areas.

## Request Branching

VISION:Builder enables you to select records and parts of records according to criteria specified on the request statement. This is accomplished by means of the logical expressions explained previously in this chapter. If the conditions specified in the logical expressions are met, the expression is true; if the conditions are not met, the expression is false. If conditional branching operators are not used following logical expressions, VISION:Builder controls the subsequent request processing flow. If the expression is true, the next statement in the request is executed. If the expression is false, control is immediately passed to end of request.

Branch operators, when specified in a request statement, together with their respective branch locations, conditionally or unconditionally transfer control from the current statement in a request to another statement in the same request or another request.

- Conditional branch is taken depending on the outcome of the preceding relational/logical expression.
- Unconditional branch is taken every time the statement is executed.

If preceded by a logical expression, that expression must be true or control is passed to end of request and the statement containing the unconditional branch is not executed.

**Note:** Refer to the *VISION:Builder Reference Summary* for details of each specification.

**Note:** Branch statements that branch within a request are always forward.

**Note:** Branch statements that branch to another request can be forward or backward. See Back Branching on page 11-13 for more information about backward branching.

**Note:** Type 1 and 2 requests or procedures are described in Chapter 15, *Advanced Transaction Processing*.

The branch operators and their locations are shown in the following table.

| Branch Operator | Branch Locator |
|---|---|
| NS (Conditional) | SEQUENCE NUMBER |
| GS (Conditional) | END |
| GO (Unconditional) | OUTPUT |
| | NEXT REQUEST |
| | NEXT MASTER |
| | REQUEST request name |
| | SUB request name |
| | SUB set name |
| | RETURN |
| | TYPE 1 |
| | TYPE 2 |

## Conditional Branching

The NS operator causes a conditional transfer to a branch location when the preceding logical expression is false. When the expression is true, the NS operation is bypassed and the next statement is executed.

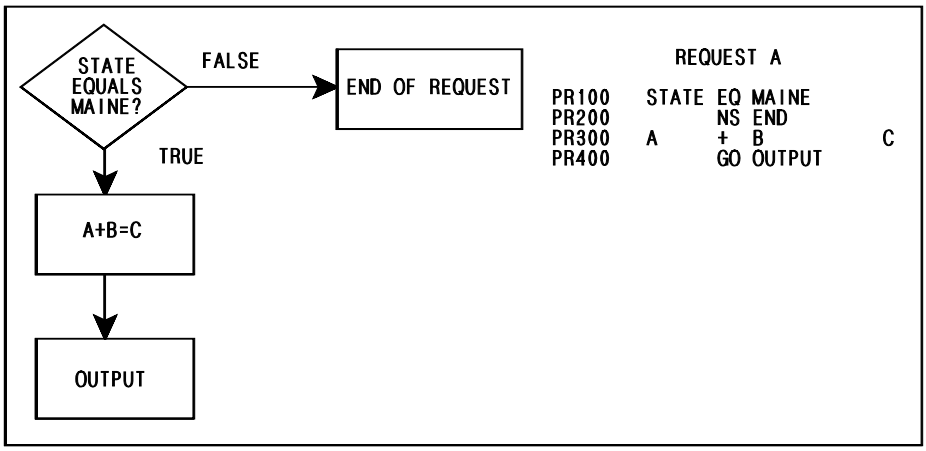



Figure 11-3 Conditional Branching Using NS

The GS operator causes a conditional transfer to a branch location when the preceding logical expression is true. When the expression is false, the GS operation is bypassed and the next statement is executed.

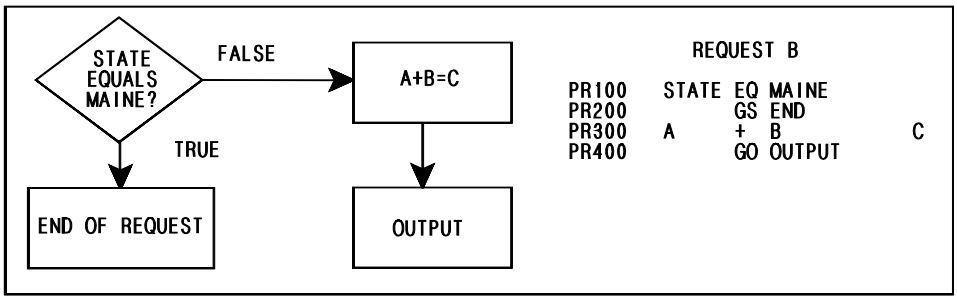



Figure 11-4 Conditional Branching Using GS

If a logical expression is not followed by a conditional branch, the flow of the request is identical to the flow when an NS END follows the logical expression. The flow of Figure 11-5 is the same as the flow of Figure 11-3.

```
REQUEST C

PR100   STATE EQ MAINE
PR200   A     +  B          C
PR300         GO OUTPUT
```

Figure 11-5 Branching Without a Conditional Branch

# Branch Locations

## Within a Request

- Sequence Number causes a branch to the request statement in that request with the specified sequence number (always higher than the request statement containing the branching operation).
- END causes a branch to end of request. Additional data cells in an array and additional segment occurrences will be processed by the request looping as described in Chapter 8, *Data Structure Handling*. Exit is positive or negative depending on the last test.
- OUTPUT causes a branch to the report/subfile sections of that request, bypassing subsequent request statements. Exit is positive.

### Beyond a Request

- NEXT REQUEST causes an immediate branch to the next request by terminating all processing of segment occurrences (or data cells in the array) in the current request including output. The current record is available to the next request.
- NEXT MASTER stops all further processing of the current master record and bypasses processing of this master record for all following requests when the branch is executed in a type N request. When executed in any request type, all following requests of the same type are bypassed. Processing resumes at the next point, following the last request of that type in the application run flow. A subroutine returns control to the calling routine.
- REQUEST Request Name causes a branch to the specified request. All processing in the current request stops for the current record and the current record is available to the specified request. The specified request can be any request in the run, including the same request that executes the branch. The request must be of the same type, except for transaction requests❹ that can branch to type 1 requests. This method of backward branching in VISION:Builder is explained in Back Branching on page 11-13.
- SUB Request Name causes a branch to the subroutine request. Subroutine requests are described in Subroutine Requests and Subroutine Sets on page 11-11.
- SUB Set Name causes a branch to a subroutine request set. Subroutine request sets are described in Subroutine Requests and Subroutine Sets on page 11-11.
- RETURN, when issued in a subroutine (request or set), terminates processing the current record (stops looping) in the subroutine (request or set) and transfers control to the calling request. See Chapter 8, *Data Structure Handling* for details on subroutine looping.

  When issued in any other request type, all processing for all requests of the current request type is terminated. VISION:Builder continues processing as if there were a GO NEXT REQUEST operation in the last request of that request type.
- TYPE 1❹ causes an immediate branch to the first type 1 request. See Chapter 15, *Advanced Transaction Processing*.
- TYPE 2❹ causes an immediate branch to the first type 2 request. See Chapter 15, *Advanced Transaction Processing*.

Branch operations are useful for controlling VISION:Builder automatic looping through segment occurrences during processing of an application request.

Figure 11-6 illustrates branching within a request. The request is divided into sections as described in Request Sections on page 11-1. It is important to note that branching within a request does not terminate looping on a lower level segment repetition. This is described in detail in Chapter 8, *Data Structure Handling*.

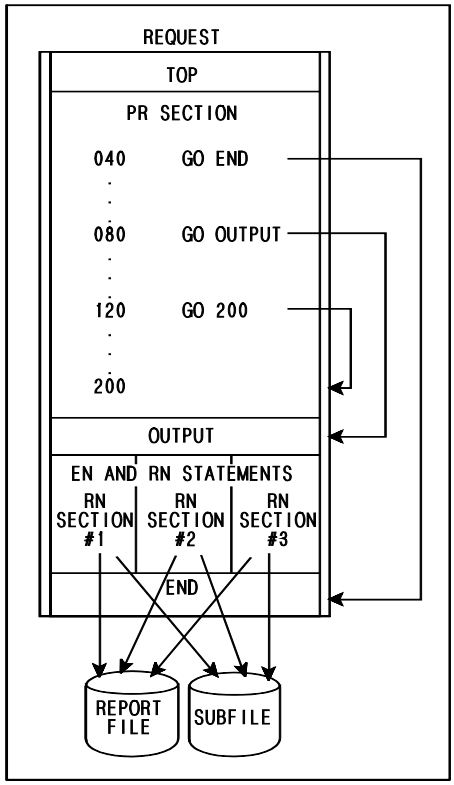



Figure 11-6 Branching Within a Request

Figure 11-7 and Figure 11-8 illustrate branching beyond a request. Figure 11-7 shows next request, next master, and request name branching. Note that the two request branches are to the top of a request; this is always the case. All three of these branch locations terminate looping on lower level segments. See Chapter 8, *Data Structure Handling*.

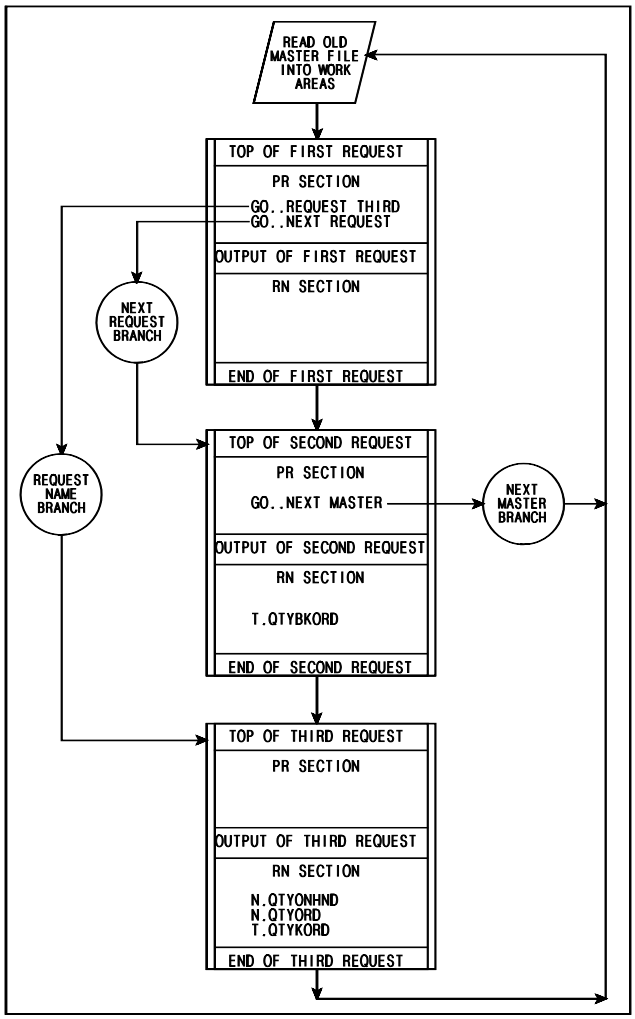



Figure 11-7 Branching Beyond a Request

Figure 11-8 shows branching to a subroutine set, an individual subroutine, and return. When the branch is to a subroutine set, all requests within that set are processed before returning to the calling request, unless an explicit return branch is executed. The return is always to the statement following the original branch in the calling request. Any individual subroutine request can be accessed from a calling request; it may or may not be part of a subroutine set.

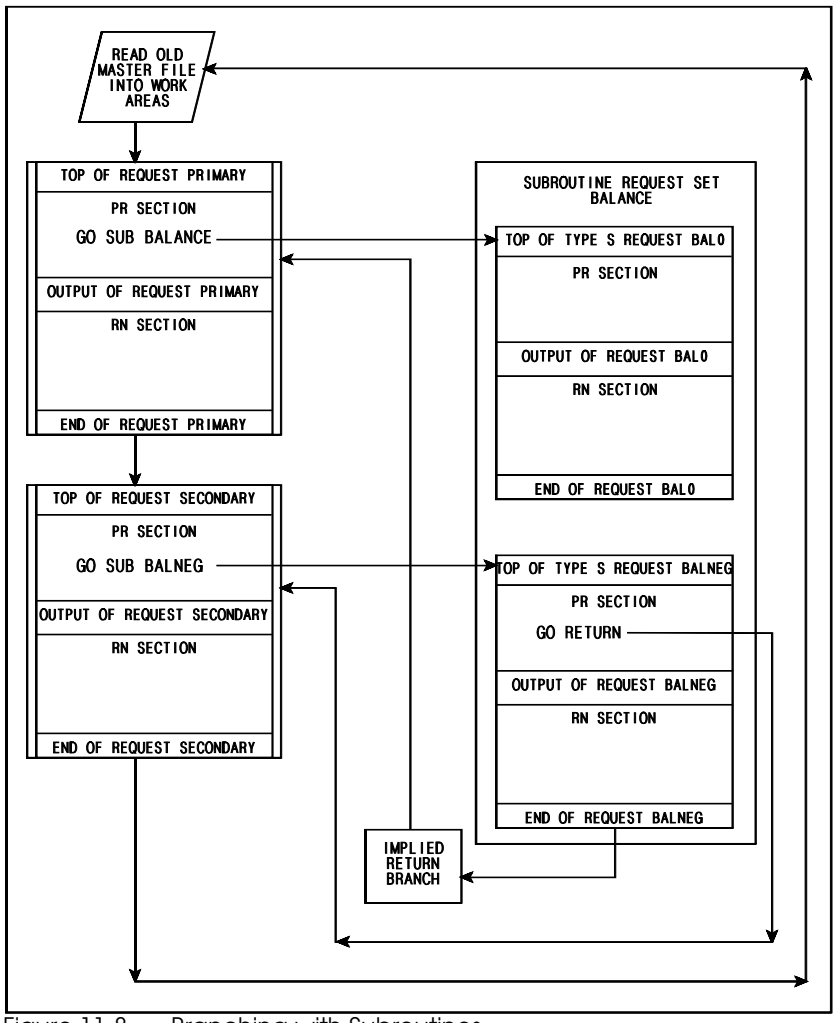



Figure 11-8 Branching with Subroutines

# Subroutine Requests and Subroutine Sets

Subroutine requests can be called by any other requests to perform a set of functions and return to the request that called them.

Several subroutine requests can be grouped with a common subroutine set name. This group of subroutine requests is called a subroutine request set.

## Subroutine Request

An individual subroutine request is a VISION:Builder request that is identified as a subroutine by entering REQUEST in the Type entry in the Application Definition window. Such a request is not invoked automatically by VISION:Builder; it can be

invoked only from another request by using a branching operator. Subroutine request sets are not available in VISION:Workbench for DOS. To code a subroutine request set, use the ER statement, positions [57-64].

A subroutine request is invoked by using any branching operator with a branch field consisting of the subroutine request name preceded by the prefix SUB and a blank (such as GO SUB request-name, GS SUB request-name, or NS SUB request-name). An individual subroutine request can be invoked by any number of requests within a run and by any request type, including another subroutine request. However, recursive invocation of a subroutine request is not permitted (that is, no subroutine request can be specified that could invoke itself either directly or indirectly). For example, subroutine request A cannot invoke subroutine request B if subroutine request B could also invoke subroutine request A.

Normal termination of a subroutine request causes processing to continue with the operation immediately following the branching operation that invoked the subroutine request with the GO, GS, or NS operations.

Figure 11-9 shows the flow of control when branching to a subroutine request B1. The calling request is REQUEST1.

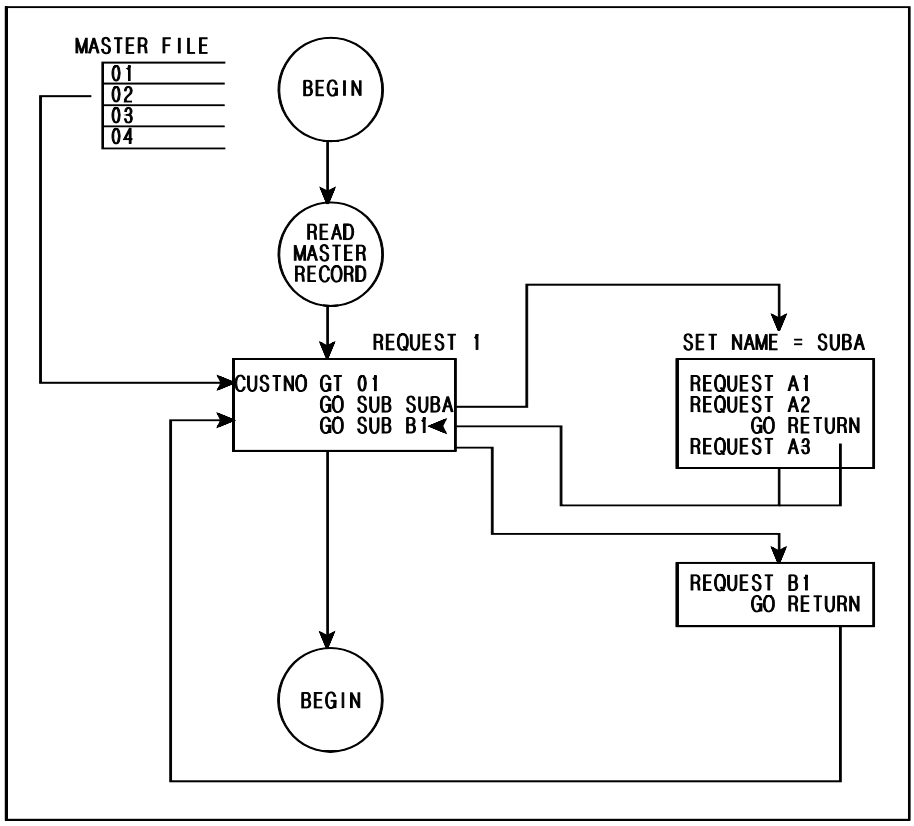



Figure 11-9 Branching to a Subroutine Request Set

### Subroutine Request Sets

A subroutine request set is composed of one or more subroutine requests, each having the same subroutine set name. The subroutine set name is entered in the set name specification of an ER statement. Invoking a subroutine set causes all individual subroutine requests within the set to be invoked in a sequential fashion based on their order within the subroutine set.

The requests that make up the subroutine set must be grouped together within the input job stream. If they are not grouped together, they are seen as two separate subroutine sets with the same name and a diagnostic message is issued.

Subroutine sets can be invoked only from another request by using the branching operators GO, GS, and NS in the form GO SUB set-name, GS SUB set-name, or NS SUB set-name. Subroutine sets can be invoked by any number and any type of requests, including a subroutine request. However, recursive invocation of subroutine sets is not permitted.

Normal termination of a subroutine request that is invoked as a member of a set causes processing to continue with the next subroutine request in the set. Normal termination of the last request in the set causes processing to continue with the operation immediately following the branching operation which invoked the set. Control is always returned to the invoking request whenever a GO RETURN, GS RETURN, NS RETURN, GO NEXT MASTER, GS NEXT MASTER, or NS NEXT MASTER operation is encountered.

Subroutine request names and subroutine set names must be unique among themselves (that is, a subroutine request and a subroutine set cannot have the same name). However, they can duplicate non-subroutine request names (that is, a subroutine set and type E request can have the same name).

Figure 11-9 shows the flow of control when branching to a subroutine request set SUBA. Control is passed to the first subroutine request A1 in the set and then to A2. The GO RETURN branch causes control to return to the statement following the GO SUB SUBA branch in the calling request, REQUEST1. If the GO RETURN branch is bypassed, control falls to the next subroutine request in the set, request A3.

## Back Branching

The back branching capability allows you to branch to either:

- A prior request of the same type in the application run, except for transaction requests.❹ See Chapter 15, *Advanced Transaction Processing*.
- The beginning of the current request.

Figure 11-10 shows how, in either case, the current record being processed is repeatedly passed back until the maximum number of back branch executions allowed is reached. The Back Branch Ctl entry on the Request Definition window

or on the ER statement indicates the number of back branch executions allowed. The back branch control entry is required to eliminate the possibility of getting into an endless program loop.

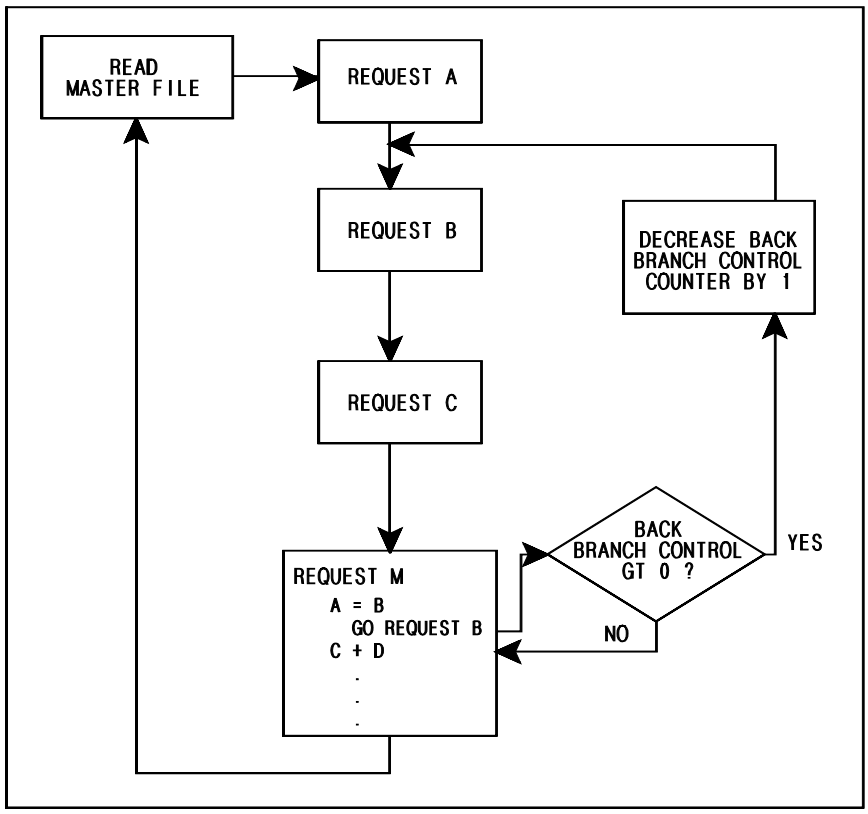



Figure 11-10 Back Branching Flow

Requests A through M must be the same request type, except for transaction requests.❹ See Chapter 15, *Advanced Transaction Processing*. Back branch control regulates the total number of back branches from this request while processing one record through this series of requests of the same type. If there is more than one back branch (to any previous request) in request M, the back branch control counter would be decreased by one every time either back branch was executed. If back branch control in request M is left blank, no back branches are executed.

Figure 11-11 and Figure 11-12 illustrate back branching in a subroutine request.

This example computes interest on overdue payments at a rate of 1.5 percent per month. Figure 11-11 shows a sample record.

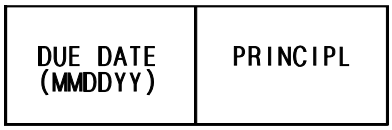



Figure 11-11 Back Branching Sample Record

Figure 11-12 shows how to solve the problem by using back branching in a subroutine request to compute the interest amount. The request DATEAMT computes the number of months over which interest is charged. The called subroutine request INTEREST loops to compute the actual interest amount.

```
                                                  ****************************
                                                  *  REQUEST NAME - DATEAMT  *
                                                  *  INPUT STREAM REQUEST    *
                                                  ****************************
-------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                                 MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ         BACK BR RNT AB      SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID           ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM      73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(ER)                                                                                (N)                                          12
-------------------------------------------------------------------------------------------------------------------------------
STMT SEQ  LOG CON .....OPERAND-A..... OPER  ............OPERAND-B.............  ......RESULT...... PARTIAL FIELD  SEQUENCE   LIST
TYPE NO. LEV CTR QLF FIELD   SEG-LVL  ATN   QLF FIELD   SEG-LVL  (OR CONSTANT)   QLF FIELD  SEG-LVL  STRT NUMB OPR  73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(PR) (010)         (F,TODAY   )        (R )                                      (T,YY      )         ( 5) ( 2) (A)              13
(PR) (020)         (T,YY      )        (- )     (DUEDATE ) 01-1                  (T,YY      )         ( 5) ( 2) (B)              14
(PR) (030)         (T,YY      )        (LT)   (D,0                            )                                                  15
(PR) (040)                              (NS)     (060                           )                                                  16
(PR) (050)                              (GO)     (NEXT REQUEST                  )                                                  17
(PR) (060)         (T,YY      )        (* )   (D,12                           ) (T,YY      )                                      18
(PR) (070)         (F,TODAY   )        (R )                                      (T,MM      )         ( 1) ( 2) (A)              19
(PR) (080)         (T,MM      )        (- )     (DUEDATE ) 01-1                  (T,MM      )         ( 1) ( 2) (B)              20
(PR) (090)         (T,MM      )        (+ )   (T,YY      )                       (T,LOOP    )                                      21
(PR) (100)                              (R )     (PRINCIPL) 01-1                  (T,NEWAMT  )                                      22
(PR) (110)                              (GO)     (SUB INTEREST                  )                                                  23
(PR) (120)         (T,NEWAMT  )        (- )     (PRINCIPL) 01-1                  (T,INTEREST)                                      24

                                                  ****************************
                                                  *  REQUEST NAME - INTEREST *
                                                  *  INPUT STREAM REQUEST    *
                                                  ****************************
-------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                                 MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ         BACK BR RNT AB      SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID           ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM      73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(ER)                                                                                (S)          ( 1200)                         25
-------------------------------------------------------------------------------------------------------------------------------
STMT SEQ  LOG CON .....OPERAND-A..... OPER  ............OPERAND-B.............  ...... RESULT.....  PARTIAL FIELD  SEQUENCE   LIST
TYPE NO.  LEV CTR QLF FIELD   SEG-LVL ATN   QLF FIELD   SEG-LVL  (OR CONSTANT)   QLF FIELD  SEG-LVL  STRT NUMB OPR  73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(PR) (010)         (T,LOOP    )        (GT)   (D,0                            )                                                  26
(PR) (020)         (T,NEWAMT  )        (* )   (D,1.015                        ) (T,NEWAMT  )                                      27
(PR) (030)         (T,LOOP    )        (- )   (D,1                            ) (T,LOOP    )                                      28
(PR) (040)                              (GO)     (REQUEST INTEREST              )                                                  29
```

Figure 11-12 Back Branching Traditional Syntax Listing

# Requests and ASL Procedures

ASL (Advanced Syntax Language) can be used in place of the fixed format VISION:Builder syntax for writing procedural logic (PR statements). When ASL is used, the term procedure is used in place of request to identify this component of the application. Procedures, unlike requests, cannot have embedded specifications for reports and subfiles. Reports and subfiles that depend upon some conditional and procedural logic must be defined as subroutines and called using the ASL CALL statement. Refer to the *VISION:Builder ASL Reference Guide* for a full description of the ASL language syntax.

A procedure is assigned a type just as with requests (see Request Types on page 11-4). A major distinction, however, is that there is no way to branch from one procedure to another procedure (even of the same type) other than by specifying a subroutine procedure by a CALL statement. The use of ASL procedures requires that the application (or that part of the application that is coded in ASL) be structured such that the flow of control through a series of subroutine procedures

be controlled by a single controlling procedure of each type. Although the use of ASL constrains the design of an application in this way, this structured approach is the preferred way even if the entire application is coded using fixed format VISION:Builder requests.

The equivalent of back branching (looping) in an ASL procedure is accomplished by use of the DO loop constructs (DO FOR, DO WHILE, DO UNTIL, and DO FORALL or combinations thereof). With these constructs, any problem that would otherwise be solved using request back branching can just as readily be solved with ASL and the appropriate looping construct. Note that the body of the looping construct (all of the statements between the DO statement and the corresponding END statement) is equivalent to a subroutine request with the appropriate back branching controls.

Figure 11-13 shows the same example as shown in Figure 11-12 but using an ASL procedure in place of the fixed format procedural syntax.

```
                                          ****************************
                                          *  PROC NAME - SAMPLE      *
                                          *  INPUT STREAM PROCEDURE  *
                                          ****************************
-------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                              MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ         BACK BR RNT AB      SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID        ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM     73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(ER)                                                                          (N)                                          4
     ***** ASL Procedure Follows - Using Translator Version x.xx
     ##PROC                                                                                                                5
     IF N.NEWFLDA EQ 'ACME INC.'                                                                                           6
       LET N.NEWFLDB = 'REPORTED BY SAMPLE PROCEDURE'                                                                      7
       LET T.COUNTER = T.COUNTER + 1                                                                                       8
       IF T.COUNTER LE 5                                                                                                   9
         CALL SAMPREPS                                                                                                    10
       ELSE                                                                                                               11
         LET PF(F.EOF 1 1) = 'E'                                                                                          12
       END                                                                                                                13
     END                                                                                                                  14
     ##PEND                                                                                                               15
     ***** End of ASL Procedure

                                          ****************************
                                          *  REQUEST NAME - SAMPREPS *
                                          *  INPUT STREAM REQUEST    *
                                          ****************************
-------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                              MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ         BACK BR RNT AB      SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID        ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM     73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(ER)                                                                          (S)                                         16
-------------------------------------------------------------------------------------------------------------------------------
        SP  Q              E  N  S  D  C  S  T  C  C  M  M  A  G  D Q PCT/RAT     P                PARTIAL   SEQUENCE   LIST
STMT SEQ BF  L  FIELD   SEG-  N  P  E  E  T  U  0  U  N  A  I  V  M  E L  FIELD   RA- C      PICTURE    STRT NO   73 TO 80    SEQ
TYPE NO. COL F  NAME     LVL  D  R  Q  S  L  B  T  M  T  X  N  G  R  C F  NAME    TIO T       EDIT      CHR CHR
-------------------------------------------------------------------------------------------------------------------------------
(R1)        (N,NEWFLDA ) 01-1                                                                                              17
(R1)        (N,NEWFLDB ) 01-1                                                                                              18
(R2)        (T,COUNTER )                                                                                                   19
(R2)        (F,EOF     )                                                                                                   20
(R3)        (N,NEWFLDA ) 01-1                                                                                              21
(R3)        (N,NEWFLDB ) 01-1                                                                                              22
(R3)        (T,COUNTER )                                                                                                   23
(R3)        (F,EOF     )                                                                                                   24
```

Figure 11-13 Sample ASL Procedure and Request in Traditional Syntax

Chapter

# 12 Tables

In VISION:Builder, the term table is used to describe a list of information that consists of two elements: a code (argument) and its expanded meaning (result). The table is searched by matching a data value (input argument) against the codes (arguments) stored in the table. When a match is found, the expanded meaning (result) stored in the table is extracted. This is called a table lookup in VISION:Builder.

VISION:Builder tables are easy to define and use. They provide extensive capabilities beyond seeking simple match conditions. You can also retrieve results searching for a table argument value that is nearest, larger, or smaller than your input argument. Additionally, a result can be returned that is calculated from the result values of two consecutive table arguments between which an input argument falls.

All tables used by VISION:Builder applications must be predefined and cataloged in the common library. The table definition consists of a description of the table's argument field and result field characteristics, the actual argument and result values, and the type of table to be used. Once the table is defined and cataloged into a VISION:Builder common library, it is available to all applications that might require the information.

There are three types of tables used in VISION:Builder. Tables can be binary, sequential, or displacement. The table type determines the lookup search technique that is used to retrieve results.

There are two methods that can be used to retrieve information from VISION:Builder tables. You can perform table lookups procedurally in your application procedures and requests using the appropriate command or operator, or you can perform table lookups automatically by predefining table result fields in file and array definitions.

The input arguments used in table lookup functions can be any field in any work area. Variable length fields cannot be used as an input argument for table lookups.

When a table lookup is performed and no matching value can be found in the table, the field designated as the result field for the lookup is flagged as invalid.

This chapter discusses the three table types, their respective search techniques, how to define VISION:Builder tables, and how to perform table lookups.

# Table Types

The three table types available in VISION:Builder are binary, sequential, and displacement. Each table type has its own search technique that is used when a table lookup is performed against the table. Depending on the information volume and the information content to be stored in the table, along with how you will access the information, each table type has its own distinct advantages.

## Binary Table

Binary tables are the most commonly used table type. VISION:Builder automatically orders and stores binary table entries by their argument values during the definition/maintenance run. Binary tables are searched using a standard binary search technique. This technique compares an input argument against the ordered table arguments, starting at the midpoint of the table, to locate a match or a subsection of the table that might contain a match. Subsequent comparisons are done against the midpoints of decreasingly smaller subsections of the table until a match is found or no more subsections exist.

Definition/maintenance can be performed on the entries in a binary table to delete and add information, affecting only the entries referenced. A new entry is automatically added in the proper ordered position based on its argument value. The maximum number of entries allowed in a binary table is 32,767.

A binary table is the default table type in VISION:Builder. Binary tables are the most versatile, accommodating both the simple equal and not equal search technique and the more advanced search techniques of nearest, next larger, and next smaller. You can also have results returned that are calculated using linear interpolation of two table result values based on the relative position of the input argument between two table arguments. Binary tables are most efficient when a large number of entries are accessed in an unpredictable order.

The following table shows you the maximum number of compares required to lookup an equal entry in a binary table based on the number of entries in the binary table

| Number of Table Entries | Maximum Compares |
|---|---|
| 10 | 4 |
| 100 | 6 |
| 500 | 9 |
| 1000 | 10 |
| 5000 | 13 |

The following table is an example of how the binary search technique would locate an equal entry in a binary table. The binary table has ten entries and the input argument for the lookup matches the ninth entry. Only three compares are needed to get a result.

| Given the Input Argument '90' | | |
|---|---|---|
| **Argument** | **Result** | **Compares** |
| 10 | Ten | |
| 20 | Twenty | |
| 30 | Thirty | |
| 40 | Forty | |
| 50 | Fifty | First try |
| 60 | Sixty | |
| 70 | Seventy | Second try |
| 80 | Eighty | |
| 90 | Ninety | Third try and found |
| 100 | One Hundred | |

## Sequential Table

Sequential table entries are stored in the physical order that the argument values are entered by the user and input to the definition/maintenance run. VISION:Builder does not automatically order sequential table entries. Sequential tables are searched one entry at a time, serially, in physical order. The input argument is compared against the first table argument, then the second, and so on, until a match is found or all arguments are checked.

Sequential tables are typically used when there is a small number of entries or the first few entries are known to be the most commonly accessed. Because VISION:Builder does not order sequential table arguments, you should order the entries with the most frequently used entries being specified first to ensure processing efficiency.

Definition/maintenance can be performed on the entries in a sequential table to delete and add information, affecting only the entries referenced. A new entry is added at the end of the table. The maximum number of entries allowed in a sequential table is 32,767.

The following table is an example of how the sequential search technique would locate an equal entry in a sequential table. The sequential table has ten entries and the input argument for the lookup matches the ninth entry. With a sequential search technique, this requires nine compares to get a result

| Given the Input Argument '90' | | |
|---|---|---|
| **Argument** | **Result** | **Compares** |
| 10 | Ten | First try |
| 20 | Twenty | Second try |
| 30 | Thirty | Third try |
| 40 | Forty | Fourth try |
| 50 | Fifty | Fifth try |
| 60 | Sixty | Sixth try |
| 70 | Seventy | Seventh try |
| 80 | Eighty | Eighth try |
| 90 | Ninety | Ninth try and found |
| 100 | One Hundred | |

## Displacement Table

Displacement table entries consist of table results that are stored in slots whose relative position is based on the numeric value of its table argument. Only table result values are stored in a displacement table. The search technique is to directly retrieve a result using an input argument. The numeric value of the input argument is used to compute the relative position of a result within the table.

Displacement table arguments are always integer numbers. The numbers should be consecutive. If consecutive numbers are not used, VISION:Builder fills in the missing numbers, creates a dummy table result at the relative position of the table, and flags it as unused. Retrieving an unused table result entry causes the designated table lookup result field to be flagged as invalid.

Definition/maintenance can be performed on the entries in a displacement table to delete and add information, affecting only the entries referenced. Deleted entries are converted to dummy entries and flagged as unused. A new entry is added at its relative slot in the table or it can replace a dummy entry. The maximum size of a displacement table is 32,000 bytes of information. The size is computed as the length of the table result field multiplied by the number of entries in the table.

The displacement search technique is the most efficient method available. Displacement tables are useful in handling blocks of data that can be referenced numerically. However, the restrictions of numeric only arguments and the physical table size limit (especially when non-consecutive entries are used) require careful review when using this table type.

The following table is an example of a displacement table. The table argument is a month number and the result is the month name. The search for an equal entry is direct; the month value is used to locate the entry at a relative position within the table. If you had an input argument of 09, the result SEPTEMBER would be retrieved directly from the table; a single search is all that is done.

| Given the Input Argument '09' | | |
|---|---|---|
| **Argument** | **Result** | **Compares** |
| 0 | ** unused ** | |
| 1 | January | |
| 2 | February | |
| 3 | March | |
| 4 | April | |
| 5 | May | |
| 6 | June | |
| 7 | July | |
| 8 | August | |
| 9 | September | Single try and directly found |
| 10 | October | |
| 11 | November | |
| 12 | December | |

Note that the table argument values are not physically stored as part of the table. They are only used to determine the relative slot for storing the table result value.

# Defining Tables

All VISION:Builder tables must be defined and cataloged in a common library before they can be referenced in an application. They must also be cataloged before they can be referenced by a file or array definition when an automatic table result field is specified.

Tables are defined in VISION:Workbench for DOS using the Table Definition window. In the Table Definition window, you enter the information describing the table: the Table Name and Table Type. Next, the characteristics for the table argument and table result fields are entered. Finally, you enter the actual table argument values along with their corresponding table result value. The table definition statements, TB and TE, are used to define tables using fixed format syntax. Figure 12-1 shows a Table Definition window for a binary table.

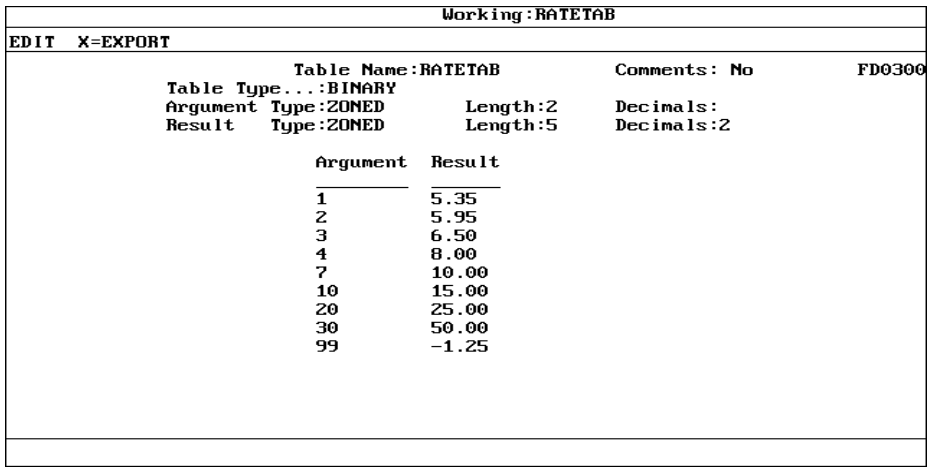



Figure 12-1 Table Definition Window of Binary Table

The RATETAB table is a BINARY table type. Note that if you define decimal places for either the table argument or table result fields, you must enter an explicit decimal so VISION:Builder can properly align the data values. If no decimal point is entered, the numeric value is handled as an integer. Negative values are specified by entering a leading minus sign.

Once you have defined and cataloged your tables, you are ready to perform table lookup functions.

# Table Lookup Functions

Tables are easy to use in VISION:Builder. There are two methods that you can use to lookup (retrieve) information in a VISION:Builder table. You can perform an automatic table lookup or a procedural table lookup.

## Automatic Table Lookup

Automatic table lookup functions are performed by referencing a field in a file or array work area that has been defined as a table result field in the file or array definition. The table result field referenced in a procedure or request statement causes a table lookup function to be performed automatically. The referenced table result field receives the result value from the lookup. To make this happen, all you need to do is predefine that table result field in your file or array definition.

Figure 12-2 shows a Segment Definition window that contains the definition of a table result field.

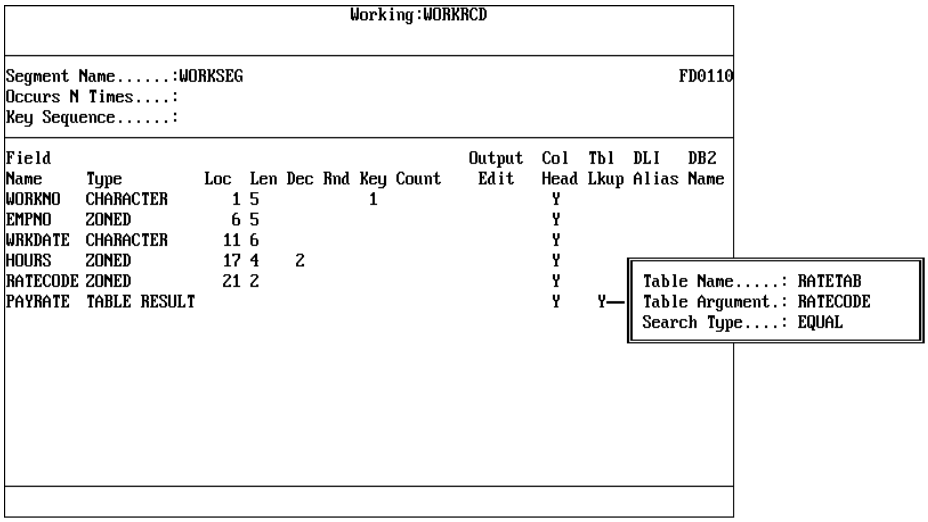



Figure 12-2 Segment Definition Window for a Table Result Field

In this example, you have a root segment named WORKSEG in a file definition named WORKRCD. The field called PAYRATE is the TABLE RESULT field type. By selecting the Tbl Lkup entry for this field, a popup window appears. Enter the table name, table argument, and search type information.

The table name must be a table that has already been defined. This example uses the RATETAB shown in Figure 12-1. The table argument (RATECODE) can be any field in the file definition except a variable length field or another table result field. The Search Type entry specifies how you want the lookup search to be performed, in this case, an EQUAL search. Notice that you do not need to specify a Loc or Len entry for a Table Result type field. The field characteristics are the same as the result field in the specified table (RATETAB). There is no Loc value because a Table Result field does not physically exist in the record. It only exists in the application work area for the file.

To summarize, when you reference field PAYRATE in an application procedure or request, the data value in the field RATECODE is used as the input argument in a lookup function to retrieve a result from table RATETAB. The table result, based on an EQUAL match to a table argument, is placed in the PAYRATE field in the file work area. The value in PAYRATE can be used in processing operations or output just like any other field in an application.

## Procedural Table Lookup

Procedural table lookups are performed by including the appropriate command or operator in an application procedure or request. The execution of the command or operator causes VISION:Builder to perform the specified lookup.

The LOOKUP function is specified in an ASL command to initiate the table lookup in a procedure. The table must have already been defined. The input argument can be a field from any file work area, a temporary field, or a system flag field. The search types include equal, nearest, smaller, larger, and interpolate. In fixed format syntax, the operators are TL (equal), TN (nearest), TS (smaller), TB (larger), and TI (interpolate).

Input argument fields need not be of the same length or data type or have the same number of decimal places as the table's argument specified in the table definition. If the input argument and table argument field attributes do not agree, VISION:Builder automatically converts the input argument to the appropriate format. Input argument fields used with displacement tables can be of any data type, but only the integer portion is used when table lookup is performed.

The result field need not have the same attributes as the table result field. VISION:Builder automatically converts the table result value to the attributes of the result field. The result field can be a field in any work area available to your application. If the result field is a temporary field that was not previously defined, it is assigned the attributes of the table result field at the time the LOOKUP function is performed.

The result field can be used just like any other field for record selection, processing, or report and subfile output. If an input argument field value is not found in the table, the result field is treated as an invalid field and is handled by VISION:Builder in the same manner as any other invalid field. See Chapter 4, *Field Characteristics and Handling* for details on field conversions and invalid fields.

## Standard Search for Equal Table Arguments

The following is an example of a table lookup that searches the table arguments for a value that equals the input argument and retrieves the corresponding table result. Figure 12-3 shows a table called SALESTAX that contains county names and their corresponding sales tax amounts.

```
                                    Working:SALESTAX
EDIT  X=EXPORT
                        Table Name:SALESTAX          Comments: No          FD0300
              Table Type...:BINARY
              Argument Type:CHARACTER   Length:15    Decimals:
              Result   Type:ZONED       Length:5     Decimals:4

                       Argument         Result
                       ________         ______
                       KERN             .0700
                       LOS ANGELES      .0825
                       ORANGE           .0775
                       SAN DIEGO        .0775
                       VENTURA          .0725
```

Figure 12-3 Table Definition Window of SALESTAX Table

To retrieve a value from the SALESTAX table, using a field called COUNTY in a master file, you would code the following command.

```
LET T.RATE = LOOKUP(SALESTAX  COUNTY)
```

Because you did not specify a search type, the search of the SALESTAX table is performed looking for an EQUAL table argument value to match the input argument value in the field COUNTY. If the COUNTY field contained a value of LOS ANGELES, the result value .0825 from the table would be placed into the T.RATE field.

If the field T.RATE was not previously defined, VISION:Builder creates a five-byte zoned field with four decimal places using the attributes of the table's result field. If the field T.RATE was already defined, the table result value converts to the format of the T.RATE field. If T.RATE was defined as having only two decimal places, the result is .08, truncating the last two decimal digits.

Remember that with equal searches, if no matching table argument is found, the result field is flagged as invalid.

## Advanced Search Types

With VISION:Builder, there are four additional search types that can be used to retrieve a result from a table. These advanced search types can only be used against binary tables. Remember that binary table arguments are in ascending order by their value.

The four advanced search types are:

- Retrieve a result from an equal or the nearest (in value) table argument entry.
- Retrieve a result from an equal or the next smaller (lower in value) table argument entry.
- Retrieve a result from an equal or the next larger (higher in value) table argument entry.
- Derive the result by linear interpolation.

When the nearest or interpolation search type is used, the table arguments in the table must be numeric.

When searching using an advanced search type, the search checks for an equal first. If no equal table argument is found, the table argument that satisfies the search type criteria is located. The nearest table argument is the one that has the smallest numerical difference with the input argument. If the input argument is exactly halfway between two table arguments, the table argument with the higher value is chosen as the match. The next smaller table argument is the one that is the next lower entry sequentially in the table. The next larger table argument is the one that is the next higher entry sequentially in the table.

### Nearest Search Type

The first example illustrates a table lookup using the nearest search type. Figure 12-4 shows a table called THOUSAND. This table rounds a series of four digit numbers (integers) to the nearest even thousand value.

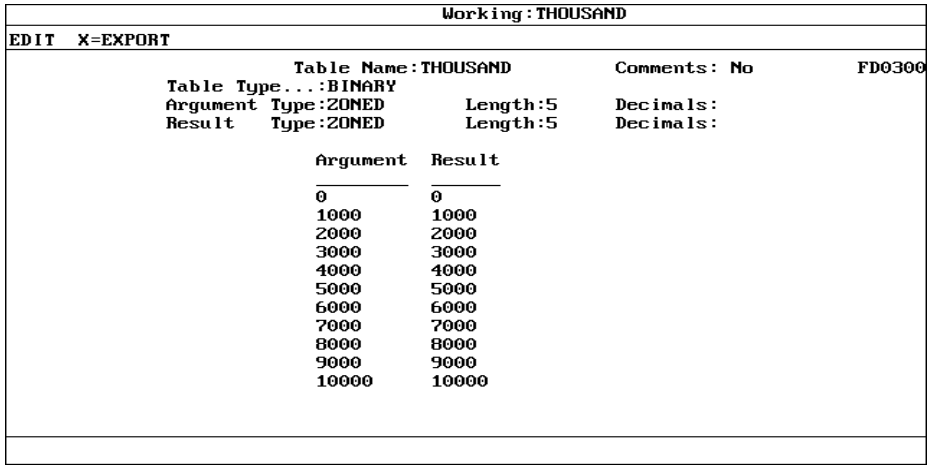



Figure 12-4 Table Definition Window of THOUSAND Table

Once the four digit number is input to the application, the following command performs the lookup function:

```
LET T.RESULT = LOOKUP(THOUSAND  T.NUMBER  NEAREST)
```

The input argument T.NUMBER is matched against an equal or the nearest table argument and a table result value is placed in the T.RESULT field. The following table shows a series of input arguments that when matched against the THOUSAND table arguments cause the indicated table result values to be retrieved

| Input Argument | Matched Table Argument | Retrieved Table Result |
|---|---|---|
| 50 | 0 | 0 |
| 999 | 1000 | 1000 |
| 1575 | 2000 | 2000 |
| 3000 | 3000 | 3000 |
| 3001 | 3000 | 3000 |
| 4275 | 4000 | 4000 |
| 5499 | 5000 | 5000 |

### Larger Search Type

This example illustrates a table lookup using the larger search type. Figure 12-5 shows the same table as Figure 12-4. The THOUSAND table rounds the same series of four digit numbers (integers) to the next larger even thousand value.

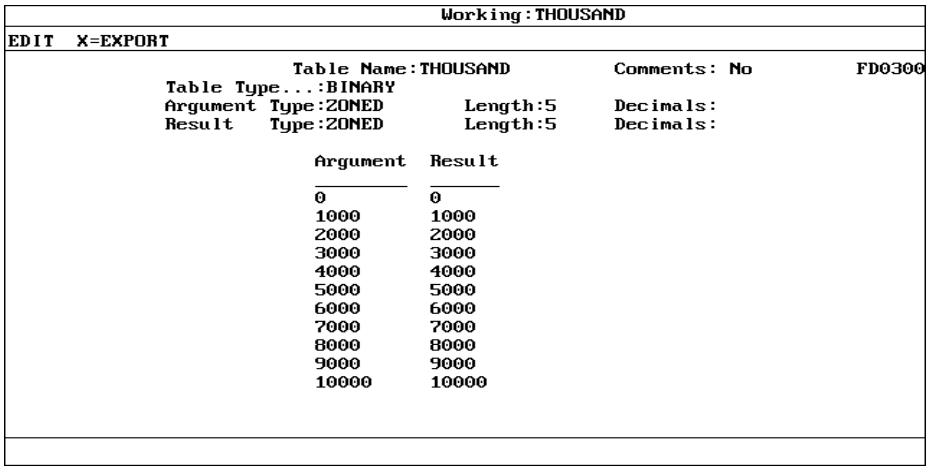



Figure 12-5 Table Definition Window of THOUSAND Table

Once the four digit number is input to the application, the following command performs the lookup function:

```
LET T.RESULT = LOOKUP(THOUSAND  T.NUMBER  LARGER)
```

The input argument T.NUMBER value is matched against an equal or the next larger table argument and a table result value is placed in the T.RESULT field. The following table shows a series of input arguments that when matched against the THOUSAND table arguments cause the indicated table result values to be retrieved.

| Input Argument | Matched Table Argument | Retrieved Table Result |
|---|---|---|
| 50 | 1000 | 1000 |
| 999 | 1000 | 1000 |
| 1575 | 2000 | 2000 |
| 3000 | 3000 | 3000 |
| 3001 | 4000 | 4000 |
| 4275 | 5000 | 5000 |
| 5499 | 6000 | 6000 |

### Smaller Search Type

This example illustrates a table lookup using the smaller search type. Figure 12-6 shows a table called JULIANDD. This table determines the Gregorian month number for the Julian day.

```
                                    Working:JULIANDD
EDIT  X=EXPORT
                        Table Name:JULIANDD          Comments: No          FD0300
              Table Type...:BINARY
              Argument Type:ZONED        Length:3     Decimals:
              Result   Type:CHARACTER    Length:8     Decimals:

                         Argument  Result
                         ________  ______
                         000       01JAN000
                         032       02FEB031
                         060       03MAR059
                         091       04APR090
                         121       05MAY120
                         152       06JUN151
                         182       07JUL181
                         213       08AUG212
                         244       09SEP243
                         274       10OCT273
                         305       11NOV305
                         335       12DEC334
                         366       ********
```

Figure 12-6 Table Definition Window of JULIANDD Table

Once the three digit Julian day is number input to the application, the following command performs the lookup function:

```
LET  T.MMCCCDDD = LOOKUP(JULIANDD  T.DAYNUM  SMALLER)
```

The input argument T.DAYNUM value is matched against an equal or the next smaller table argument and a table result value is placed in the T.MMCCCDDD field. If the input day number was 182, the retrieved value is 07JUL181. If the input day was 210, the retrieved value is 07JUL181. Any input day with a value greater than 366 retrieves the value ********. This is in the table to indicate that the input day is not valid (that is, greater than 365).

This table retrieves several pieces of information about the input Julian day with a single lookup function. The result field T.MMCCCDDD for the table lookup contains eight characters of information. By using the partial field (PF) function you can extract the portion of the retrieved information that you would need to complete the processing algorithm.

The first two characters in the T.MMCCCDDD field are the Gregorian month, in number form, that the Julian day represents. The third through fifth characters are the Gregorian month in abbreviated character form. Finally, the last three positions (6 through 8) of T.MMCCCDDD contain a value that can be used to compute the Gregorian day within the month, subtracting this value from the input Julian day produces the Gregorian day within the month. This example shows that the result information stored in a table can be used to contain more than one piece of data for a given argument. The result must be characters so the partial field function can be used. For more information about partial fields, see Chapter 4, *Field Characteristics and Handling*.

### Interpolation Search Type

The interpolation search type is actually a lookup and calculation function. The input argument is used to search the table arguments and locate an equal table argument value or two consecutive table argument values that bracket the input argument. When the equal table argument is found, the table result is retrieved and no calculation is needed. However, when no equal table argument is found, a bracket is established. The table result values associated with the bracket are used to calculate a final result value. The final result is calculated using linear interpolation. The following table shows the formula that is used to calculate the final result

| Final Result | = | (X - AL) * (RH - RL) / (AH - AL) + RL |
|---|---|---|
| X | = | Input Argument Value |
| AL | = | Table Argument - Lower Bracket Value |
| AH | = | Table Argument - Higher Bracket Value |
| RL | = | Table Result - Lower Bracket Value |
| RH | = | Table Result - Higher Bracket Value |

The following table illustrates the interpolation search type and how the bracketing occurs. Notice that when the input argument is numerically compared to the table arguments, it is positioned between an argument that is lower in value and one that is higher in value.

| Search Position | Table Argument | Table Result |
|---|---|---|
| | " | |
| | " | |
| | " | |
| | AL | RL |
| X | | |
| | AH | RH |
| | " | |
| | " | |
| | " | |

Once the bracket is established, the final result can be calculated using the linear interpolation formula.

Figure 12-7 shows a table of costs based on a quantity ordered. The table is called CARPETS.

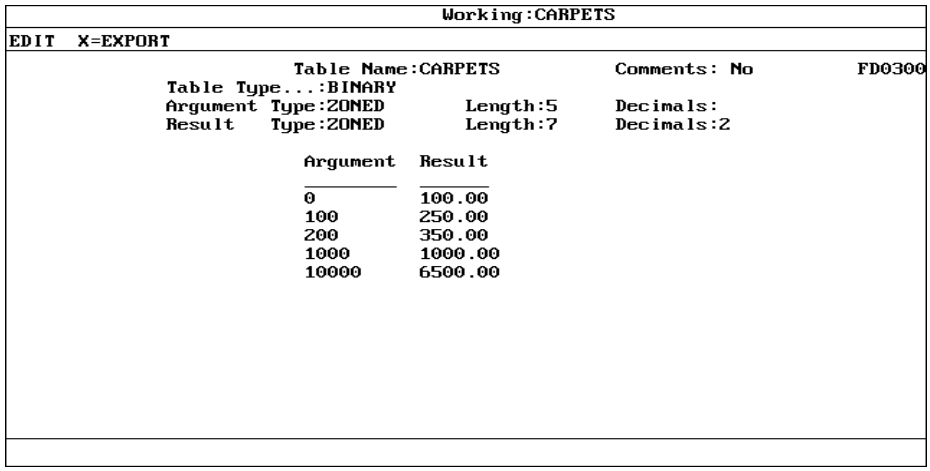



Figure 12-7 Table Definition Window of CARPETS Table

This example calculates the cost of carpet per square foot. The price per square foot varies depending on the quantity ordered (that is, the more that is ordered, the less the cost per square foot). The CARPETS table has the price breaks at the various quantities (square feet). The interpolation search calculates the exact cost based on the amount of carpet, in square feet, that is ordered.

Once the square feet ordered quantity is input to the application, the following command performs the table lookup:

```
LET T.COST = LOOKUP(CARPETS  T.SQFEET  INTERPOLATE)
```

The input argument T.SQFEET value is matched against the CARPETS table argument values. If a matching table argument value is found, the corresponding table result value is placed in the lookup result field T.COST and no further calculation is needed. However, if the input argument value in T.SQFEET falls between two table argument values, the linear interpolation calculation is performed to obtain a result that can be placed into the lookup result field T.COST.

The following table shows a series of input arguments, the bracketed table arguments, and the calculated results when the interpolation search type is used against the CARPETS table

| Input Argument | Lower Table | | Higher Table | | |
|---|---|---|---|---|---|
| Value (X) | Argument (AL) | Result (RL) | Argument (AH) | Result (RH) | Calculated Result |
| 250 | 200 | 350.00 | 1000 | 1000.00 | 390.63 |
| 150 | 100 | 250.00 | 200 | 350.00 | 300.00 |
| 500 | 200 | 350.00 | 1000 | 1000.00 | 593.85 |
| 999 | 200 | 350.00 | 1000 | 1000.00 | 999.19 |
| 10 | 0 | 100.00 | 100 | 250.00 | 115.00 |
| 3750 | 1000 | 1000.00 | 10000 | 6500.00 | 2680.56 |

VISION:Builder automatically applies the linear interpolation formula in the table starting on page 12-13 and returns the calculated result. The calculated result is automatically rounded (half adjusted) before it is placed into the lookup result field.

## Out of Range Input Argument Values

The last thing to consider regarding the advanced search types is when the input arguments for the lookup function are outside the range of values in the table. The following table shows the results of a lookup function if the input argument value is less than all the table argument values or greater than all the table argument values. When the result field is flagged as invalid, the general rules of VISION:Builder invalid fields apply.

| | Result of Lookup if Input Argument is: | |
|---|---|---|
| Search Type | Below Range of Table | Above Range of Table |
| Nearest | Lowest Entry | Highest Entry |
| Smaller | * (invalid) | Highest Entry |
| Larger | Lowest Entry | * (invalid) |
| Interpolate | * (invalid) | * (invalid) |

Chapter

# 13 Arrays

In addition to hierarchical record structures, VISION:Builder also supports array record structures. Arrays can organize, summarize, and store large amounts of information. The information stored in an array record is easily accessed for processing, reporting, and outputs. Array records can be stored in files and used as input to or output from VISION:Builder applications.

**Note:** This chapter contains examples to illustrate and enhance the concepts and facilities that are described. These examples are portions of complete job streams. For the complete versions of the examples, Appendix D, *Complete Sample Applications*.

The following members are referenced in this chapter:

| Job Name | Member Name | Description |
|---|---|---|
| Preparation job | R13PREPA | The Preparation job defines and builds the files and data sets used in the applications that are the examples. |
| Examples job | R13FIGSA | The Examples job contains applications that demonstrate the coding and outputs described and shown with this chapter. |

The actual JCL, source code and data for these job streams are delivered on the VISION:Builder system tape in the Examples file. This file is downloaded during the installation process. See the *VISION:Builder Installation Guide* for more information.

An array record structure is organized along two dimensions known as rows and columns. At the intersection of each row and column there exists a data cell. A data cell consists of one or more data fields. Each data cell in an array record contains the same set of fields (that is, a single occurrence of the data cell fields). The array record contains multiple occurrences of the data cells equal to the number of rows times the number of columns defined for the array record. Additionally, when array records are used in input or output files, the array record also has a single key field that uniquely identifies the entire array record.

VISION:Builder provides you with the capability to define, manipulate, and process these array record structures. The definition of an array is simple. You specify the array dimensions (rows and columns), the data cell fields and their characteristics, and, optionally, the array record key. Commands, operators, and automatic processing mechanisms are provided to reference and manipulate all the data cells in an array record. You can also limit the array processing to a group of data cells, a specific row, or a specific column within an array record. Finally, you can limit the processing to a specific data cell (the intersection of a row and column within the array record). Arrays can be built and processed entirely in memory and/or they can be handled as input and output records.

Within the array record structure, a data cell occurrence is always uniquely identified by its specific row and column number. Indexing the data cells within an array record is performed by row and column number reference. VISION:Builder provides a facility called array looping that automatically makes each data cell occurrence available for processing by referring to a data cell field by name. Array looping can be performed on an entire array record or a selected section (a row or a column or a data cell).

For example, an array record consists of two rows and three columns. Each data cell contains two fields, FIELDA and FIELDB. Figure 13-1 shows how the array record is structured in memory.

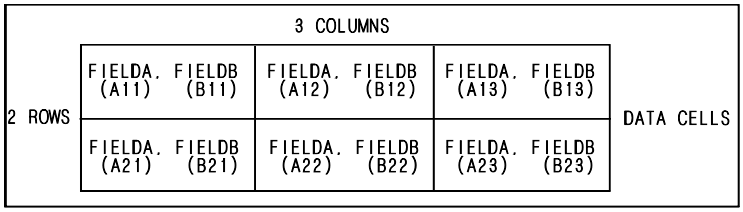



Figure 13-1 An Array Record in Memory

In Figure 13-1, the data values are shown for the fields in each data cell. The data value is the field letter A or B followed by the relative row and column number. This is done to give you an idea of how the data cells are indexed. VISION:Builder has two system flags named ROW and COLUMN that always contain the relative numbers of the data cell occurrence processing. In Figure 13-1, the first data cell is row one, column one; the last data cell is row two, column three.

When an array record is stored on a file, its format consists of a record key field followed by all the data cell occurrences. The data cell occurrences are concatenated in numerical order by their relative row and column numbers. Figure 13-2 shows how the array record structure in Figure 13-1 is formatted on a file.

| | DATA CELL | DATA CELL | DATA CELL | DATA CELL | DATA CELL | DATA CELL |
|---|---|---|---|---|---|---|
| KEY FIELD | FIELDA, FIELDB (A11) (B11) | FIELDA, FIELDB (A12) (B12) | FIELDA, FIELDB (A13) (B13) | FIELDA, FIELDB (A21) (B21) | FIELDA, FIELDB (A22) (B22) | FIELDA, FIELDB (A23) (B23) |

Figure 13-2 An Array Record on a File

Notice that the key field is the first field in the record and is not part of any data cell.

When arrays are input and processed as an array, they are input as a coordinated input file. Normally when array records are output to a file, the entire array record is selected to the subfile. Coordinated input files are described in Chapter 6, *Input File Processing* and subfiles are described in Chapter 7, *Output File Processing*.

This chapter describes how to define arrays, how array looping functions, and various ways you can process arrays. An application example illustrates the definition and manipulation of an array.

It is important to note that array record structures and data record structures are closely related and conceptually the same. The main difference is that an array record structure has only a single path that consists of a root segment and a fixed occurring level two segment type (the data cell occurrences). Whereas, data record structures can have many segment types and paths. Also, separate commands and operators are used to control the looping in array structures versus data structures. See Chapter 8, *Data Structure Handling* for more information on data record structures.

# Array Definitions

The definition of array structures is very similar to the definition of data structures. In the array definition, you specify the details that describe the characteristics of your array, including the array dimensions (rows and columns) and data cell field details. The glossary listing of your array definition shows the details of your array specifications. The beginning of the glossary shows the array name, the characteristics of its records, and the number of rows and columns. The remainder of the glossary shows the characteristics of the array data cell fields and, optionally, an array record key field.

Figure 13-3 shows an Array Definition Glossary Listing and the definition of an array called COSTARR.

```
                                    DETAILED GLOSSARY BY LOCATION FOR                                  GLOSSARY PAGE      1
                                       ARRAY DEFINITION - COSTARR
DATE CATALOGED - 11/05/01 16:56:48                                    DATE LAST UPDATED - 11/05/01 16:56:48
CATALOGED BY -                                                        LAST UPDATED BY -
EXPIRATION DATE -                                                     DATE LAST USED -
NUMBER OF ROWS = 52                                                   RECORD FORMAT = KEY SEQUENCED VSAM
NUMBER OF COLUMNS = 8                                                 RECORD SIZE = 7492              (COMPUTED)
DATA CELL SIZE = 18                                                   BUFFER SIZE = 7492
ARRAY SIZE = 7492
-------------------------------------------------------------------------------------------------------------------------
FIELD     FIELD    FIELD  FIELD  FIELD    DEC     ARRAY     EDIT CODES     EDIT    OUTPUT   LINE       REPORT FIELD
NAME     LOCATION  LENGTH TYPE   RNDING  PLACES    KEY      ( ) ( ) ( )   LENGTH   WIDTH     NO   ***     HEADING    ***
-------------------------------------------------------------------------------------------------------------------------
YEAR         1       4     C                       1                        4        4
                                                                                           1   ***      YEAR      ***
                                                                                           2   ***      NO.       ***
-------------------------------------------------------------------------------------------------------------------------
WEEKNO       1       2     C                                                2        4
                                                                                           1   ***      WEEK      ***
                                                                                           2   ***      NO.       ***
DAYNO        3       1     C                                                1        3
                                                                                           1   ***      DAY       ***
                                                                                           2   ***      NO.       ***
```

Figure 13-3 Array Definition Glossary Listing

```
HOURS        4      5     Z              1                          8       8
                                                                                  1   ***     HOURS      ***
                                                                                  2   ***     WORKED     ***
COSTS        9      5     Z              1                          8       8
                                                                                  1   ***     COSTS      ***
                                                                                  2   ***     1000'S     ***
SALES       14      5     Z              1                          8       8
                                                                                  1   ***     SALES      ***
                                                                                  2   ***     1000'S     ***
** CLSAG00  TYPE 0  END OF ARRAY DEFINITION GLOSSARY.                                                                  *****
```

Figure 13-3 Array Definition Glossary Listing

The COSTARR array stores information on costs and sales, for a year, by week and day. The dimensions are 52 rows for the weeks and eight columns for the days (Monday through Sunday and a weekly total). The array record is a key sequenced (VSAM KSDS) format and the buffer size has been optimized. The size of an array record is calculated as the key field length, plus the sum of the data cell field lengths multiplied by the number of rows and columns. The size entries are optional and are computed automatically by VISION:Builder.

The array definition includes a key field. The key field is only needed if you plan to use this array definition with an external array, an array that is input and/or output from an application and the access method for the file requires a key. The array key may be necessary when reading in an array record as a coordinated input file or if the array record is output to a subfile. Arrays being built and processed in memory only do not require a key. Coordinated input files are described in Chapter 6, *Input File Processing* and subfiles are described in Chapter 7, *Output File Processing*. Remember, the key field is not part of any data cell and only identifies the entire array record. In the example, the field YEAR is the array key field.

Next, you specify the set of fields that exist in each data cell occurrence within the array record. WEEKNO, DAYNO, HOURS, COSTS, and SALES are the data cell fields. Column headings (not shown) are specified for each field. These are used when reporting the data from the array. You can also specify table result fields in an array. Table result fields are described in Chapter 12, *Tables*.

The array definitions follow the same rules as file definitions. The following are some specifics that pertain to arrays.

- The array record key, if defined, must be defined starting in location one.
- Only fields in one data cell occurrence are defined starting at location one.
- The last byte of the last data cell field establishes the size of a data cell.
- Variable length fields are not allowed.
- The largest size for an entire array record is 16,777,215 bytes.
- If changes are made to an existing array definition that is already cataloged in a common library, the entire array definition must be re-cataloged.

Once you define the array and catalog it in the VISION:Builder common library using the usual definition/maintenance run, you are ready to use the array in an application.

To define an array using fixed format syntax coding statements, use the array definition statements AD, B0, Bn, and BA. For more information, refer to the *VISION:Builder Reference Summary*.

# Array Looping

The power of VISION:Builder and its method for processing array record structures is the concept called array looping. VISION:Builder provides commands, operators, and automatic mechanisms that make the processing of array record structures a simple task. Once you understand the essential concept of array looping and how it can be controlled, application development becomes easy.

The array looping mechanism automatically makes every array data cell occurrence available to your application for processing and outputs. You can automatically access every array record data cell occurrence, a portion of the data cell occurrences (one row or one column), or just a single array record data cell occurrence.

The following sections describe automatic array looping and the various methods in which you can control array looping to meet specific application requirements.

Remember, array structure looping and data structure looping are essentially the same process.

## Simple Array Looping

The first executed reference, within a procedure or a request, to a data cell field in an array record starts an automatic loop through all the data cell occurrences in the array record work area. The loop is automatically generated and controlled by VISION:Builder. Each data cell occurrence in the array record is made available for processing and/or output.

The array loop starts with the data cell occurrence at the intersection of row one and column one. The loop proceeds to the next data cell occurrence in the row (that is, row one and column two). When all the data cell occurrences for a row (all the columns in the row) are available to the application, the array looping process proceeds to the first data cell occurrence in the next row (that is, row two and column one). The array looping process continues until the data cell occurrence of the last row and last column in the array record is available to the application.

The purpose of simple array looping is to make all the data in all the data cell occurrences of an array record available for processing and output using a single executed field reference.

Figure 13-4 illustrates simple array looping. To report all the data from an array, reference an array data cell field (FIELDA or FIELDB) in a report output request and the simple array looping mechanism makes all the data cell occurrences available for the report output.

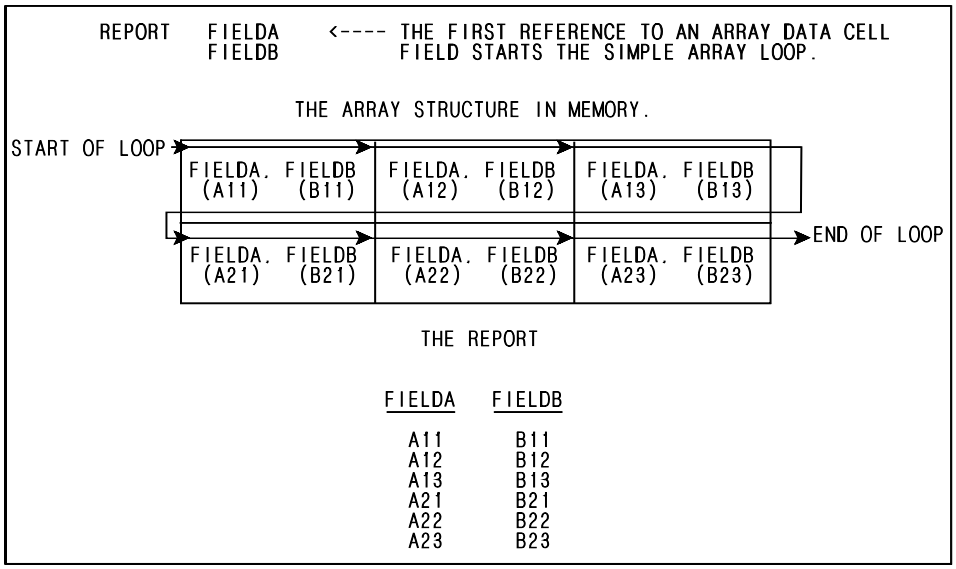



Figure 13-4 Simple Array Looping Illustration

## Nested Array Looping

Nested array looping can occur when references to data cell fields from more than one array record are executed in the same procedure and/or request for processing and/or output. A simple array loop is established for each array record referenced. The simple array loops are nested so that the array loop on the second referenced array record occurs within the array loop of the first referenced array record. An array loop on a third referenced array record occurs within the second array record loop and so on. When this nesting of array loops occurs, the inner most array record loop (the last referenced array) is completely executed before the next outer array record loop proceeds to its next data cell occurrence. This inner to outer execution of array loops continues until the outer most array record loop (first referenced array) is completed.

Figure 13-5 illustrates the effect of nested array looping when two arrays are referenced.

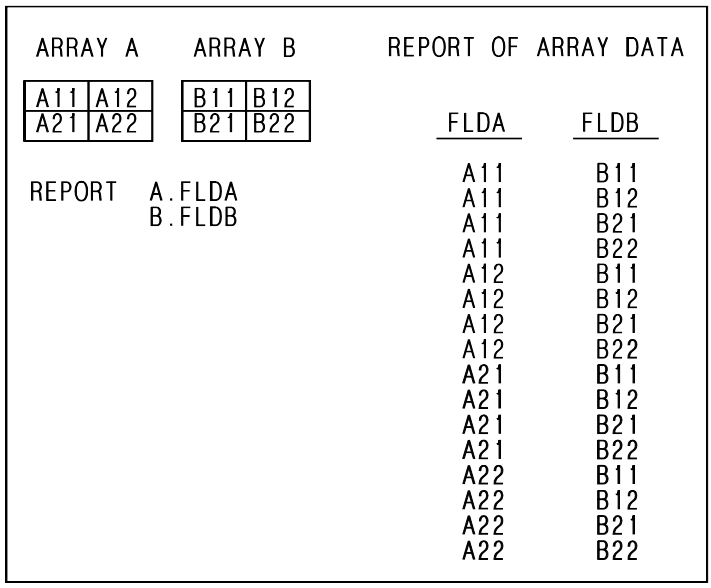



Figure 13-5 Nested Array Looping Illustration

You have two arrays known as Array A and Array B. Each is defined as two rows by two columns and each data cell occurrence contains just one data field. You want to report the data from each array. In a report output request, you reference a field from Array A and a field from Array B. Two simple array loops are started, one in each array. For each data cell made available by the simple array loop on Array A, the simple loop on Array B is started, processed, and completed. This means that for each Array A data cell occurrence that is made available, *every* Array B data cell occurrence is made available. This nest of simple array loops continues until all Array A data cell occurrences are made available for the report output. The array report output in Figure 13-5 shows the results of the nested array looping.

## Parallel Array Looping

As you have seen in the previous section, nested array looping can occur when executed references to more than one array record exist in the same procedure and/or request. However, you may not want to report all the permutations that are made available by nested array looping.

When performing report or subfile outputs of array data from an application, the nested array looping can be suppressed. The simple array loops in each referenced array can be processed in parallel instead of nested. Parallel array looping causes the array looping mechanism to simultaneously make the next data cell occurrence from all referenced arrays available for output.

Figure 13-6 illustrates the effects of parallel array looping when two arrays are referenced.

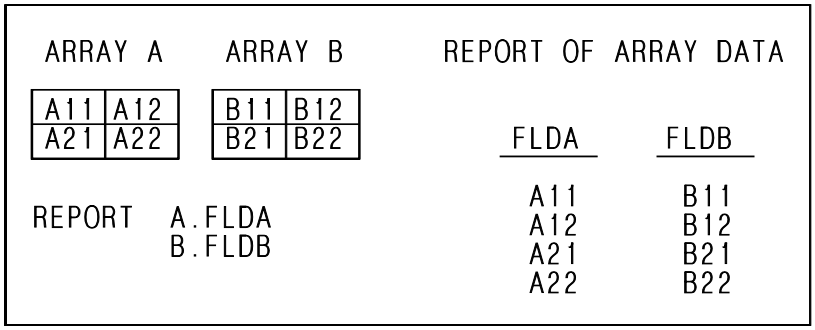



Figure 13-6 Parallel Array Looping Illustration

You have two arrays known as Array A and Array B. Each is defined as two rows by two columns and each data cell occurrence contains just one data field. You want to report the data from each array. In a report output request, you reference a field from Array A and then a field from Array B.

To use parallel array looping instead of the nested array looping, specify to VISION:Builder that you want the report produced using the Abbreviated Format.

See Figure 13-7 for an example of the Abbreviated Format specification on the Report command.

```
[report-name:]  REPORT field-name...,
      ABBREVIATED
```

Figure 13-7 Report Command Showing the Abbreviated Format Keyword to Activate Parallel Array Looping on Report Output

Simple array loops are started, one in each array. A data cell occurrence from each array is made available for output. Once the output is performed, the simple array loops on both arrays advance to the next data cell occurrence in each array, in parallel. This parallel array looping movement continues through the arrays until all data cell occurrences from both arrays are made available for the report output. The array report output in Figure 13-6 shows the results of the parallel array looping process.

To perform parallel array looping on subfile output from two or more arrays, specify the ABBREVIATED keyword on the EXTRACT command. Figure 13-8 shows an example of the EXTRACT command.

```
[extract-name:] EXTRACT FILE subfile-name,
     field-name ...,
     ABBREVIATED
```

Figure 13-8 EXTRACT Command Showing the ABBREVIATED Keyword that Activates Parallel Array Looping for Subfile Outputs

## Controlling Array Looping

You can control the array looping mechanism using the following methods:

- Array Operations
- Selection Control
- Branching
- Subroutine Processing

These methods are described in detail in this section. The techniques for controlling array looping are the same as those for controlling data structure looping, only the array operators are unique to arrays. Chapter 8, *Data Structure Handling* discusses automatic looping on data structures and how it is controlled.

### Array Operations

The LOCATE command controls array looping, indicates specific array operations, and limits the array looping mechanism to selected portions of the array record. You can limit the array looping to a single row, to a single column, or to a single data cell occurrence.

In the LOCATE command, identify which array you want to control by specifying the array qualifier on the Files Used window.

Next, identify which portion of the array you want to limit the array looping process to by specifying the row and/or column number.

- Specify a row number only to limit the array loop to the data cell occurrences in the specified row.
- Specify a column number only to limit the array loop to the data cell occurrences in the specified column.
- Specify both a row and column number to limit the array loop to a single data cell occurrence (that is, no array looping takes place).

Once you have specified a LOCATE command, the array looping mechanism is limited until the current procedure or request is complete. To remove the array loop limitations specified by a LOCATE command, you can execute a RELEASE command identifying the array. Once the RELEASE command is executed, simple array looping can occur or another LOCATE command can be executed.

The system flag ASTATUS contains a value indicating the status of the last array operation (LOCATE command) executed in your procedure or request.

The following are some examples of using the LOCATE command. Figure 13-9 shows an array with two rows and three columns. It is assigned the qualifier A.

| | | |
|---|---|---|
| FIELDA. FIELDB<br>(A11) (B11) | FIELDA. FIELDB<br>(A12) (B12) | FIELDA. FIELDB<br>(A13) (B13) |
| FIELDA. FIELDB<br>(A21) (B21) | FIELDA. FIELDB<br>(A22) (B22) | FIELDA. FIELDB<br>(A23) (B23) |

Figure 13-9 An Array Record in Memory, Qualifier A

### Example 1 LOCATE ARRAY A ROW 1 COLUMN 3

The LOCATE command specifies a ROW and COLUMN; a single data cell occurrence is located and made available to the application. The data values A13 and B13 from FIELDA and FIELDB are available. No other data cell occurrences and data values are available.

### Example 2 LOCATE ARRAY A ROW 2

The LOCATE command specifies a ROW only; all the data cell occurrences in ROW 2 are made available to the application. A report of the data values from FIELDA and FIELDB show the values A21, B21 and A22, B22 and A23, B23. Only the data cell occurrences in ROW 2 are available.

### Example 3 LOCATE ARRAY A COLUMN 3

The LOCATE command specifies a COLUMN only; all the data cell occurrences in COLUMN 3 are made available to the application. A report of the data values from FIELDA and FIELDB show the values A13, B13 and A23, B23. Only the data cell occurrences in COLUMN 3 are available.

When specifying a ROW or COLUMN number, the value can be a constant, a field name, or an arithmetic expression whose numeric value identifies the row and/or column.

Here are some rules regarding the LOCATE command.

- Once an array loop is established, any further LOCATE commands for that array in the same procedure or request are suppressed (ignored).
- The RELEASE command terminates the hold on a single data cell that was performed by the LOCATE ROW COLUMN operation and makes the array available for subsequent array looping operations (for example, RELEASE ARRAY A).
- When a LOCATE command is unsuccessful, the array data cell fields are logically missing.

Refer to the *VISION:Builder ASL Reference Guide* for details on the LOCATE and RELEASE commands and the ASTATUS system flag.

In fixed format syntax coding, the operators used to control array looping are LA, LC, LD, and LR. Refer to the *VISION:Builder Reference Summary* for details.

## Selection Control

Selection control is another means by which you can control array looping. Sometimes, it may be necessary to loop through all the array record data cell occurrences looking for a specific value in a data cell field. Once the data cell with the specific data value is found, report the data from the single data cell occurrence without reporting data from any other data cell occurrences. When the selection logic (PR statements) and reporting output (Rn statements) are in the same request, specifying a Y in position [49] of the ER statement causes VISION:Builder to terminate the array loop once the logic is satisfied and the output is performed. The output is performed only once.

## Branching to Control Array Looping

Array looping can also be controlled with the use of VISION:Builder branching capabilities. The following table lists the effects on looping for the corresponding ASL and fixed format syntax branching locations. This is the same figure as shown in Chapter 8, *Data Structure Handling* with the comments changed to reflect the impact on array looping

| ASL Syntax | Fixed Format Syntax | Effect on Looping |
|---|---|---|
| N/A | GO NEXT REQUEST | Terminates all processing of the current request for the current record and transfers control to the next request. All looping on all array data cell occurrences in the current request stops for all arrays and the next request is processed. |
| CONTINUE | GO END | Terminates looping on the current array data cell occurrence and proceeds to the next array data cell occurrence. |
| TRANSFER NEXT MASTER | GO NEXT MASTER | Terminates all processing of the current record for all procedures and requests, including subroutine procedures. Control is transferred to the next VISION:Builder function. All looping on all arrays stops for all procedures and requests. |

| ASL Syntax | Fixed Format Syntax | Effect on Looping |
|---|---|---|
| N/A | GO REQUEST B | Terminates all processing of the current request for the current record and transfers control to request B. All looping on all array data cell occurrences in the current request stops and Request B is processed. |
| N/A | GO OUTPUT | Terminates looping on the current array data cell and proceeds to the next array data cell occurrence after output is performed. |
| RETURN or LEAVE | GO RETURN | Terminates all processing of the current subroutine procedure, subroutine request, or subroutine set. It returns to the statement following the one that called the subroutine procedure, subroutine request, or subroutine set. All array looping started in the subroutine stops. |

In addition, in fixed format syntax, a backward branch to a previous procedure or request can be taken. In this case, the effect on looping is the same as that of a forward branch (such as GO REQUEST B).

For details on fixed format VISION:Builder syntax for requests, subroutines, subroutine sets, and request branching, see Chapter 11, *Requests*. For details on branch locations in ASL procedures, refer to the *VISION:Builder ASL Reference Guide*.

## Subroutine Processing and Array Looping

Subroutine procedures and requests can be used to control array looping.

All called subroutine procedures or subroutine requests inherit any array loops that were started in the calling procedure or request. This means that references to array data cell fields in an array whose array loop was inherited have only a single data cell occurrence available. This can be very useful because any output from an array whose loop was inherited will only output data from a single data cell.

The subroutine procedures and requests cannot change any array loops that they inherit. Any LOCATE command on an array whose array loop was inherited will be suppressed and the ASTATUS system flag will be set to INHR.

However, subroutine procedures and requests can start new array loops on any array that is not currently involved (inherited) in an array loop. This is done by referencing a data cell field or executing a LOCATE command on an array that was not referenced prior to entering the subroutine procedure or request.

Any new array loops started in a subroutine procedure or request are terminated when the subroutine returns to the calling procedure or request. Newly started array loops are not passed back to the calling procedures or requests.

### Array and Automatic Looping

The automatic looping mechanism used to process data record structures and the array looping mechanism used to process array record structures can occur together in an application. When loops are activated for data record structures and array record structures, nesting of the loops can occur. The discussions in this chapter and in Chapter 8, *Data Structure Handling* are still valid because the two looping mechanisms are essentially the same. The aspects for controlling looping (parallel looping, selection control, branching, subroutines) are true even when the two looping mechanisms are operating concurrently. Just remember that the order in which the loops are activated establishes the relationship between the mechanisms (that is, outer loop to inner loop processing).

## Array Processing

Once arrays are defined to VISION:Builder and cataloged in a common library, the processing of arrays can be included in an application. Arrays are loaded with data, manipulated and processed, and output in reports or files using application procedures and requests.

Some of the advantages of processing with array record structures are:

- Arrays can be loaded in memory with data from several sources: input files, temporary fields, calculations, and others. Once loaded, the array data can be manipulated and processed in memory by procedures and requests without regard to the original source of the data.
- The information stored in the array can be summarized in memory using procedures or requests to perform complex calculations and algorithms on the rows and/or columns of data. This reduces the number of output records needed to produce reports and/or subfiles.
- Automatic array looping through an entire array (or a selected section) processes large amounts of data with just a few statements.
- Access to the array data is controlled easily by row, column, or specific data cell.

This section describes and illustrates the various processing activities of an application using arrays. This application example shows how to build (load) arrays in memory, how to manipulate the data, and how to input and output array data.

The key to array processing is the understanding of array looping. Array looping is described in detail in Array Looping on page 13-5.

## Loading an Array in Memory

Arrays can be loaded (built) with data entirely in memory using application procedures or requests. Arrays contain an array record in a work area referenced by a qualifier. This array work area is structured and easily accessed using array looping across the rows and columns.

In this example, load an array record with information about costs and sales. The information is organized by week number and day within the week. Figure 13-10 shows an Array Definition Glossary Listing and the definition of the array called COSTARR.

```
                                    DETAILED GLOSSARY BY LOCATION FOR                                  GLOSSARY PAGE      1
                                       ARRAY DEFINITION - COSTARR
DATE CATALOGED - 11/05/01 16:56:48                                    DATE LAST UPDATED - 11/05/01 16:56:48
CATALOGED BY -                                                        LAST UPDATED BY -
EXPIRATION DATE -                                                     DATE LAST USED -
NUMBER OF ROWS = 52                                                   RECORD FORMAT = KEY SEQUENCED VSAM
NUMBER OF COLUMNS = 8                                                 RECORD SIZE = 7492              (COMPUTED)
DATA CELL SIZE = 18                                                   BUFFER SIZE = 7492
ARRAY SIZE = 7492
-----------------------------------------------------------------------------------------------------------------------
FIELD     FIELD    FIELD  FIELD  FIELD    DEC     ARRAY    EDIT CODES    EDIT    OUTPUT   LINE       REPORT FIELD
NAME     LOCATION  LENGTH TYPE   RNDING  PLACES    KEY     ( ) ( ) ( )  LENGTH   WIDTH     NO   ***    HEADING    ***
-----------------------------------------------------------------------------------------------------------------------
YEAR         1       4     C                       1                       4       4
                                                                                          1   ***     YEAR      ***
                                                                                          2   ***     NO.       ***
-----------------------------------------------------------------------------------------------------------------------
WEEKNO       1       2     C                                               2       4
                                                                                          1   ***     WEEK      ***
                                                                                          2   ***     NO.       ***
DAYNO        3       1     C                                               1       3
                                                                                          1   ***     DAY       ***
                                                                                          2   ***     NO.       ***
HOURS        4       5     Z               1                               8       8
                                                                                          1   ***    HOURS      ***
                                                                                          2   ***    WORKED     ***
COSTS        9       5     Z               1                               8       8
                                                                                          1   ***    COSTS      ***
                                                                                          2   ***    1000'S     ***
SALES       14       5     Z               1                               8       8
                                                                                          1   ***    SALES      ***
                                                                                          2   ***    1000'S     ***
```

Figure 13-10 Array Definition Glossary Listing for the COSTARR Array

The COSTARR array is set up to store information on costs and sales, for a year, by week and day. The dimensions are 52 rows for the weeks and eight columns for the days (Monday through Sunday and a weekly total). The array record is a key sequenced (VSAM KSDS) format.

The array definition includes a key field called YEAR. The information stored in each data cell occurrence within the array record is WEEKNO, DAYNO, HOURS, COSTS, and SALES.

The information loaded into the COSTARR array comes from three sources: the application master file, an indexed coordinated input file, and calculations and algorithms in a procedure. The master file contains hours worked for the year by

week and day from various department stores. These amounts are loaded into the array. Figure 13-11 shows the File Definition Glossary Listing for a file called WORKHRS.

```
                                        DETAILED GLOSSARY BY LOCATION FOR                                  GLOSSARY PAGE      1
                                           FILE DEFINITION - WORKHRS
DATE CATALOGED - 11/05/01 16:56:48                                                  DATE LAST UPDATED - 11/05/01 16:56:48
CATALOGED BY -                                                                      LAST UPDATED BY -
EXPIRATION DATE -                                                                   DATE LAST USED -
FILE IDENTIFICATION =                                                               RECORD FORMAT = FIXED BLOCKED
NUMBER OF SEGMENTS IN FILE = 1                                                      RECORD SIZE = 80
NUMBER OF FIELDS IN FILE = 5                                                        BLOCK SIZE = 8000

                                        ******************************
                                        *    SEGMENT   1, LEVEL 1     *
                                        *   SEGMENT NAME = HOURS      *
                                        ******************************
SEGMENT OCCURS N TIMES = 1                                          KEY FIELD 1 = STOREID    TYPE = C   LENGTH = 5
SEGMENT SIZE = 17
NUMBER OF FIELDS IN SEGMENT = 5
-------------------------------------------------------------------------------------------------------------------------------
FIELD   FIELD   FIELD    FIELD   FIELD    DEC    CNT FIELD  EDIT CODES     EDIT    OUTPUT   LINE  DATA BASE NAME/
NAME    TYPE   LOCATION  LENGTH  RNDING  PLACES  FOR SGMT   ( ) ( ) ( )   LENGTH   WIDTH     NO   *** COLUMN HEADING ***
-------------------------------------------------------------------------------------------------------------------------------
STOREID   C        1       5                                                 5       6
                                                                                          1    ***     STORE      ***
                                                                                          2    ***     ID NO.     ***
YEAR      C        6       4                                                 4       4
                                                                                          1    ***      YEAR      ***
WEEKNO    C       10       2                                                 2       4
                                                                                          1    ***      WEEK      ***
                                                                                          2    ***      NO.       ***
DAYNO     C       12       1                                                 1       3
                                                                                          1    ***      DAY       ***
                                                                                          2    ***      NO.       ***
HRSWRKD   Z       13       5             1                                   8       8
                                                                                          1    ***     HOURS      ***
                                                                                          2    ***     WORKED     ***
```

Figure 13-11 Segment Definition Command for the WORKHRS File Definition

The application also uses an indexed coordinated input file, containing the sales information for the year by week number and day of the week. Figure 13-12 shows the File Definition Glossary Listing for the sales file called SALEAMTS.

```
                                        DETAILED GLOSSARY BY LOCATION FOR                                  GLOSSARY PAGE      1
                                           FILE DEFINITION - SALEAMTS
DATE CATALOGED - 11/05/01 16:56:48                                                  DATE LAST UPDATED - 11/05/01 16:56:48
CATALOGED BY -                                                                      LAST UPDATED BY -
EXPIRATION DATE -                                                                   DATE LAST USED -
FILE IDENTIFICATION =                                                               RECORD FORMAT = KEY SEQUENCED VSAM
NUMBER OF SEGMENTS IN FILE = 1                                                      RECORD SIZE = 50
NUMBER OF FIELDS IN FILE = 5                                                        BUFFER SIZE = 50

                                        ********************************
                                        *    SEGMENT   1, LEVEL 1     *
                                        *   SEGMENT NAME = SALES      *
                                        ********************************
SEGMENT OCCURS N TIMES = 1                                          KEY FIELD 1 = KEYFIELD   TYPE = C   LENGTH = 7
SEGMENT SIZE = 12
NUMBER OF FIELDS IN SEGMENT = 5
SEGMENT KEY CHARACTERISTICS = U
-------------------------------------------------------------------------------------------------------------------------------
FIELD   FIELD   FIELD    FIELD   FIELD    DEC    CNT FIELD  EDIT CODES     EDIT    OUTPUT   LINE  DATA BASE NAME/
NAME    TYPE   LOCATION  LENGTH  RNDING  PLACES  FOR SGMT   ( ) ( ) ( )   LENGTH   WIDTH     NO   *** COLUMN HEADING ***
-------------------------------------------------------------------------------------------------------------------------------
KEYFIELD  C        1       7                                                 7       8
                                                                                          1    ***    WEEK/DAY    ***
                                                                                          2    ***      NO.       ***
YEAR      C        1       4                                                 4       4
                                                                                          1    ***      YEAR      ***
WEEKNO    C        5       2                                                 2       4
                                                                                          1    ***      WEEK      ***
                                                                                          2    ***      NO.       ***
DAYNO     C        7       1                                                 1       3
                                                                                          1    ***      DAY       ***
                                                                                          2    ***      NO.       ***
SALES     Z        8       5            1                                    8       8
                                                                                          1    ***     SALES      ***
                                                                                          2    ***     1000'S     ***
```

Figure 13-12 Segment Definition Command for the SALEAMTS File Definition

The array application example is called ARRAYAPP. It has several procedures, several reports, and one output subfile. There are several files and definitions used by the application.

Figure 13-13 shows the Run Control commands for the ARRAYAPP application.

```
CONTROL NAME ARRAYAPP TERM LISTGEN
FILE MASTER INPUT WORKHRS KEYS NONE
FILE CORD1 SALEAMTS DIRECT BY T.INDEXKEY
FILE CORD2 COSTARR  DIRECT BY T.LASTYEAR ARRAY ; External Array Input
FILE SUBF1 ARRAYOUT                            ; Output of an Array
FILE REPORT
ARRAY A NAME COSTARR                    ; the Primary Array
ARRAY B NAME COSTARRX OVERDEFINES A     ; the Overdefined Array
```

Figure 13-13 Array Examples Application ARRAYAPP: Run Control Commands

**Note:** For a complete version of this example, see Array Examples Application ARRAYAPP: Run Control Commands – Part 1 of 10 on page D-99.

The following is a brief description of the entries on the Files Used window in Figure 13-13.

- The WORKHRS master file (M4OLD) drives the application. This file contains all the hours worked from all the various department stores in a random order. The hours are recorded by week and day number. All these records are read and made available to the application. Because this file is unsequenced, the

automatic sequence checking (KEYS NONE) is turned off. The hours worked data are loaded and summarized into the array. See Figure 13-11 for the WORKHRS file definition details.

- The SALEAMTS file is specified as an indexed coordinated input file (M4CORD1). Qualifier 1 references the work area. The direct-reads are performed using the temporary field INDEXKEY as the key of the record to be read. There is one record for each week and day in the year. The record contains the total sales amount from all department stores. These sales amounts are loaded into the array. See Figure 13-12 for the SALEAMTS file definition details.
- The COSTARR array is specified and assigned the qualifier A. The work area referenced by qualifier A is structured according to the COSTARR array definition. Because no DD-Name was specified, this is an internal array (that is, built in memory only). See Figure 13-10 for the COSTARR array definition.
- The COSTARRX entry is also an internal array and assigned the qualifier B. This array overdefines the array work area (that is, array equivalence).
- The ARRAYOUT entry specifies a subfile (M4SUBF1) for output from the application. This outputs the entire array record after it is loaded with all the information. The internal COSTARR array becomes an external array.
- The M4CORD2 entry is another indexed coordinated input file. It is an array that also uses the COSTARR array definition. This file inputs an entire array record of last year's costs and sales built by this application when it was run last year. The qualifier 2 is assigned to this external array. The direct-read is performed using the temporary field LASTYEAR as the key of the record to be read.

The MAIN procedure starts the process. Each WORKHRS master file record that is automatically read by VISION:Builder is made available to the MAIN procedure.

```
MAIN: PROC
;
; For each MASTER Work Hours Record Input
;
; Load the Internal Array
; with hours by week and day numbers
;
IF A.YEAR = '    '      ; set year in array rcd key field for
   LET A.YEAR = O.YEAR  ; we output the Array to a subfile
END IF
;
LOCATE ARRAY A ROW O.WEEKNO COLUMN O.DAYNO
;
;  (the week and day number values reference a specific
;   row & column {data cell}, no looping will occur.)
;
;     If the Week and Day Values are outside the dimensions of the
;     Array a Data Cell LOCATE Error occurs - test the ASTATUS Flag
;
```

Figure 13-14 Array Examples Application ARRAYAPP: MAIN Procedure Coding (Page 1 of 2)

```
IF F.ASTATUS = '    ' THEN              ; successful Locate
   LET A.HOURS = A.HOURS + O.HRSWRKD    ; move data into Array
   LET A.WEEKNO = O.WEEKNO
   LET A.DAYNO  = O.DAYNO
ELSE                                    ; unsuccessful Locate
   CALL REPORT HRSERR                   ; bad week or day value
END IF
;
END PROC
```

Figure 13-14 Array Examples Application ARRAYAPP: MAIN Procedure Coding (Page 2 of 2)

**Note:** For a complete version of this example, see Array Examples Application ARRAYAPP: MAIN Procedure Coding – Part 2 of 10 on page D-99.

The MAIN procedure saves the YEAR from the master file into the array key field YEAR. This is only done once when the first master file record is available. The hours worked data is stored into the array. After the data is stored, VISION:Builder reads the next master file record and makes it available to the MAIN procedure. This continues automatically until the master file is at end of file.

To locate a specific data cell in array A (COSTARR), use the LOCATE command. The objective is a single data cell occurrence at the intersection of a ROW and COLUMN identified by the numbers in the master file O.WEEKNO field and O.DAYNO field, respectively.

If the specific data cell (system flag ASTATUS is blanks) is located, load the data into the data cell's fields. The O.HRSWRKD (hours worked) data from the master file record is added to any previous amount stored in the data cell occurrence (that is, summarizing the hours worked). The O.WEEKNO and O.DAYNO values are also saved for reporting.

In the example, if the numbers in the O.WEEKNO and/or the O.DAYNO fields exceed the defined dimensions of the COSTARR array (52 and 8), the report procedure HRSERR produces an exception report.

Once you load the data into a specific data cell, you are done and you can return to the calling procedure (MAIN).

Loading data into an array, locating the appropriate data cell, and transferring the data values are simple tasks when you use the direct approach (such as a specific data cell identified by row and column numbers). You can also automatically loop through all (or a portion) of the array data cell occurrences and match data field values against array data cell field values looking for the appropriate occurrence. The usual technique is to look for a matching data value or the next unused data cell (that is, blank data cell field values). Another method converts non-numeric input data values to numbers using a table lookup. The converted values are used to directly locate a data cell occurrence.

Once all the hours worked data from the master file is loaded into the array, end of file on the hours worked master file occurs. An end of file procedure loads the sales amounts and calculates the costs for all the data cell occurrences in the array. Figure 13-15 shows the end of file procedure named EOF.

```
EOFMAIN: PROC TYPE EOF
;
; EOF Procedure - after all MASTER Work Hours Records are processed.
;
CALL PROCEDURE FILLINA  ; Calc cost, retrieve sales,
                        ; and put into the internal Array
                        ;
CALL REPORT COSTRPT1    ; Produce Report 1 (std report)
                        ; cost versus sales
                        ;
CALL REPORT COSTRPT2    ; Produce Report 2 (customized)
                        ; cost versus sales across page
                        ; using overdefined array def
                        ;
CALL SUBFILE ARROUT     ; Write the internal Array to a Subfile
                        ;
CALL PROCEDURE ARRIN    ; Read an external Array (Cord 2) for
                        ; Last Year's Data and Report the Information
;
END PROC
```

Figure 13-15 Array Examples Application ARRAYAPP: EOFMAIN Procedure Coding

**Note:** For a complete version of this example, see Array Examples Application ARRAYAPP: EOFMAIN Procedure Coding – Part 3 of 10 on page D-100.

The EOFMAIN procedure calls several subroutine procedures to manipulate, report, and output the array data. There is also a subroutine procedure that shows how you can input an array record into an application.

The first subroutine procedure (FILLINA) calculates the costs based on the hours worked and loads the sales amounts into the array data cells. Figure 13-16 shows the FILLINA procedure.

```
FILLINA: PROC TYPE SUBROUTINE

;
; Finish building the internal Array Data
; Calculate the Cost Amounts
; Retrieve Sales Data from the ICF (DIRECT) Cord File
;  and move it into the Array
; Accumulate the Weekly Sales for each row in the Array
;  and store the Amount into the Array
;
INDEXKEY: FIELD C 7    ; Temp Field for DIRECT Cord1 Read
;
; Calculate Cost in 1000's for each column (day)
; (this starts an automatic loop thru the entire Array)
;
LET A.COSTS = A.HOURS * 7.67 / 1000 WITH ROUNDING
;
; When you get to column 8 in each row, use it to store
; the Weekly Accumulated Totals from each column in      .
; the current row.
;
IF F.COLUMN = 8 THEN             ; Column 8 processing
   LET A.COSTS = T.WKCOSTS       ; Store the accum'd values
```

Figure 13-16 Array Examples Application ARRAYAPP: FILLINA Procedure (Page 1 of 2)

```
   LET A.SALES = T.WKSALES       ; from columns 1 to 7 in the row
   LET A.WEEKNO = T.WEEKNO
   LET A.DAYNO  = '8'
   LET T.WKCOSTS = 0             ; Clear accum'd values
   LET T.WKSALES = 0
;
; If not column 8, it will be column 1 thru 7
;
; Build the DIRECT Key Field to read the Sales Cord File
; Retrieve a Sales Record and move the Sales Amount into the Array
;
; Accumulate the Costs and Sales for Weekly Totals Column
;
ELSE                             ; Column 1 to 7 processing
;
  COMBINE A.YEAR A.WEEKNO A.DAYNO STORE T.INDEXKEY   ; build DIRECT Key
;
  LET A.SALES   = 1.SALES                  ; <- This causes DIRECT Read
  LET T.WKCOSTS = T.WKCOSTS + A.COSTS      ; Accum Totals
  LET T.WKSALES = T.WKSALES + A.SALES
  LET T.WEEKNO  = A.WEEKNO
;
END IF
;       Returns when all rows have been processed (Automatic Loop Ends)
;
END PROC
```

Figure 13-16 Array Examples Application ARRAYAPP: FILLINA Procedure (Page 2 of 2)

**Note:** For a complete version of this example, see Array Examples Application ARRAYAPP: FILLINA Procedure – Part 4 of 10 on page D-101.

The FILLINA procedure calculates the daily costs. This is done by multiplying the hours worked in every data cell occurrence by a pay rate and rounding the cost to thousands of dollars. The LET command does all of this in a single statement. The reference to the array data cell field A.COSTS starts an automatic simple array loop through all the data cell occurrences, one by one. Each data cell occurrence is made available to all the statements in this procedure. This finishes filling in the data in the array.

First, the A.COSTS are calculated as A.HOURS times a pay rate of 7.67, which is divided by 1000 and rounded to thousands of dollars with one decimal place.

Next, the column position is checked. The array has eight columns, one for each day (Monday to Sunday) and one for storing totals for the week (the row). The IF command checks the value in the F.COLUMN system flag to determine the position.

When you are positioned at column 8, you load the total weekly costs and sales (T.WKCOSTS and T.WKSALES) into the column 8 data cell occurrence data fields. You clear (set to zero) the temporary fields used to accumulate the weekly totals.

If you are not positioned at column 8, you perform the ELSE statements. The ELSE statements read the sales amounts from the indexed coordinated input file (M4CORD1), load the sales amount into a data cell occurrence data field, and accumulate the weekly totals.

The COMBINE command concatenates the A.YEAR, A.WEEKNO, and A.DAYNO fields from the array and current data cell occurrence and places the string into a temporary field named T.INDEXKEY. This is the value of the key of the sales record you want VISION:Builder to directly read.

The LET A.SALES = 1.SALES statement reads a sales record and transfers the data field value from the input work area transferred to the array data cell occurrence field.

Next, you accumulate the costs and sales amounts into the temporary fields T.WKCOSTS and T.WKSALES. These become the weekly totals that are stored in column 8 of each row.

This procedure is executed for each data cell occurrence in the array, automatically, by the simple array looping mechanism. When all data cell occurrences are processed (made available to this procedure), you return to the calling procedure EOF.

The array is completely loaded and filled in with all of the information and data you need for reporting and output.

## Reporting from Arrays

The reporting of array data is no different than the reporting of data from files. specify the array fields and VISION:Builder automatically loops through all the data cell occurrences and outputs the data to the report.

Continuing with the application example, you loaded and filled in the COSTARR array with costs and sales by week and day. Now you want to report the information.

The EOFMAIN procedure calls a procedure request named COSTRPT1 to produce a standard report of the information in the array. Figure 13-17 shows the REPORT command for the COSTRPT1 report procedure.

```
COSTRPT1: REPORT,
;
; Report Costs/Sales by Day and Week
;
                A.WEEKNO A.DAYNO A.COSTS A.SALES,
          TYPE SUBROUTINE
          FORMAT WIDTH 80 DATEFMT TODAYX
          TITLE 'COST AND SALES BY DAY AND WEEK'
          TITLE 'AMOUNTS SHOWN IN 1000 OF DOLLARS'
          END REPORT
```

Figure 13-17 Array Examples Application ARRAYAPP: COSTRPT1 Report Procedure

**Note:** For a complete version of this example, see Array Examples Application ARRAYAPP: COSTRPT1 Report Procedure – Part 5 of 10 on page D-102.

You specified that you want to see the week and day number, along with the costs and sales amounts. The references to the array data cell fields A.WEEKNO, A.DAYNO, A.COSTS, and A.SALES make the data from every data cell occurrence available for report output using simple array looping.

The report is automatically formatted. The field column headings from the array definition are used as the column headings above each field on the report. The amounts are edited, based on their scale. A date and page number, along with a specified report title, complete the report formatting. Figure 13-18 shows a portion of the formatted report produced from the array data by the report procedure COSTRPT1.

```
11/06/2001                COST AND SALES BY DAY AND WEEK            PAGE 1
                          AMOUNTS SHOWN IN 1000 OF DOLLARS
 ---------------------------------------------------------------------------------
  WEEK  DAY   COSTS     SALES
  NO.   NO.   1000'S    1000'S
 ---------------------------------------------------------------------------------
   01    1       2.2      10.0
   01    2       2.2      10.0
   01    3       2.2      10.0
   01    4       2.2      10.0
   01    5       2.2      10.0
   01    6       2.2      10.0
   01    7       2.2      10.0
   01    8      15.4      70.0
   02    1       2.2       9.8
   02    2       2.2       9.8
   02    3       2.2       9.8
   02    4       2.2       9.8
   02    5       2.2       9.8
   02    6       2.2       9.8
   02    7       2.2       9.8
   02    8      15.4      68.6
   03    1       2.1       9.6
   03    2       2.1       9.6
   03    3       2.1       9.6
             .
             .
             .
   51    4       3.7      16.5
   51    5       3.7      16.5
   51    6       3.7      16.5
   51    7       3.7      16.5
   51    8      25.9     115.5
   52    1       3.8      16.9
   52    2       3.8      16.9
   52    3       3.8      16.9
   52    4       3.8      16.9
   52    5       3.8      16.9
   52    6       3.8      16.9
   52    7       3.8      16.9
   52    8      26.6     118.3
```

Figure 13-18 A Portion of the Standard Report of the COSTARR Information Produced by the COSTRPT1 Report Procedure in the ARRAYAPP Application

The report shows the week and day numbers and associated costs and sales. This was produced with just a few statements.

Note that the detail lines that have a week number and a day number of 8. These are actually the weekly total data cell occurrences. These are from the column 8 data cell occurrences of each row in the array.

This report satisfies the basic requirements. However, you really visualize the data in a format that looks more like the array structure in memory. What you would like to see is a report that shows all the days of a week on one line with columns for each day and the total for the week. You would also like the names of the days shown instead of the numbers of the days.

In VISION:Builder you can easily build a different report using a different view of the array work area. Array Equivalence - Overdefining Arrays on page 13-23 shows an example of how to overdefine the array work area and produce a report that looks more like the array structure.

## Array Equivalence - Overdefining Arrays

The array equivalence mechanism in VISION:Builder allows you to overdefine your array work area in memory. The original (primary) array work array is viewed as a different (secondary) structure.

The secondary array definition is an overdefinition of the same structure and storage area defined by the primary array definition, an overlaying of a new grid of rows and columns on top of an existing grid of rows and columns. You specify a primary array qualifier and its associated secondary array qualifier and point to the same work area in memory. The following lists the rules regarding secondary array definitions and their use:

- A secondary array can only be an internal array (memory only).
- The secondary array qualifier must be an alphabetic character.
- The secondary array can only refer to one primary array.
- There can be multiple secondary arrays for the same primary array.
- The data cell size times the number of columns of the secondary array must equal the data cell size times the number of columns of the primary array. The size of a row must be equal in the secondary and primary arrays.
- The total size of the secondary array must not exceed the total size of the primary array. You can define fewer rows in the secondary array.
- Array equivalence is specified in the Run Control ARRAY command as part of the run control statements for your application (the RA statement is used for fixed format coding. See the *VISION:Builder Reference Summary* for details on the RA statement).

In the application example, ARRAYAPP, a secondary array, overdefines the primary array. The array consists of a series of rows, one for each week in the year. Each row has only one column that contains eight sets of fields for costs and sales, one for each day of the week and a weekly total. The secondary array overdefines a 52 row and 8 column array grid with a 52 row by 1 column grid. This allows you to format the report output with all the days of the week on a single detail line.

Figure 13-19 shows the Array Definition Glossary Listing for the secondary array definition, the COSTARRX array.

```
                                    DETAILED GLOSSARY BY LOCATION FOR                                    GLOSSARY PAGE      1
                                      ARRAY DEFINITION - COSTARRX

DATE CATALOGED - 11/05/01 16:56:48                                  DATE LAST UPDATED - 11/05/01 16:56:48
CATALOGED BY -                                                      LAST UPDATED BY -
EXPIRATION DATE -                                                   DATE LAST USED -
NUMBER OF ROWS = 52                                                 RECORD FORMAT = FIXED BLOCKED (DEFAULT)
NUMBER OF COLUMNS = 1                                               RECORD SIZE = 7492              (COMPUTED)
DATA CELL SIZE = 144                                                RECORDS PER BLOCK = 1           (DEFAULT)
ARRAY SIZE = 7492
-------------------------------------------------------------------------------------------------------------------------------
FIELD     FIELD    FIELD  FIELD  FIELD    DEC     ARRAY     EDIT CODES     EDIT    OUTPUT   LINE       REPORT FIELD
NAME     LOCATION  LENGTH TYPE   RNDING  PLACES    KEY     ( ) ( ) ( )   LENGTH   WIDTH     NO  ***     HEADING    ***
-------------------------------------------------------------------------------------------------------------------------------
YEAR          1      4     C                        1                      4       4
-------------------------------------------------------------------------------------------------------------------------------
WEEKNO        1      2     C                                               2       2
COSTS1        9      5     Z              1                                8       8
SALES1       14      5     Z              1                                8       8
COSTS2       27      5     Z              1                                8       8
SALES2       32      5     Z              1                                8       8
COSTS3       45      5     Z              1                                8       8
SALES3       50      5     Z              1                                8       8
COSTS4       63      5     Z              1                                8       8
SALES4       68      5     Z              1                                8       8
COSTS5       81      5     Z              1                                8       8
SALES5       86      5     Z              1                                8       8
COSTS6       99      5     Z              1                                8       8
SALES6      104      5     Z              1                                8       8
COSTS7      117      5     Z              1                                8       8
SALES7      122      5     Z              1                                8       8
COSTS8      135      5     Z              1                                8       8
SALES8      140      5     Z              1                                8       8
** CLSAG00  TYPE 0  END OF ARRAY DEFINITION GLOSSARY.                                                                    *****
```

Figure 13-19 Array Definition Glossary Listing for the Secondary Array COSTARRX Used in the ARRAYAPP Application

The COSTARRX array is defined as 52 rows and 1 column. Because it is an internal array, no other characteristics are needed.

The data cell fields are defined to map their positions in the work area as built using the primary array. The WEEKNO identifies the row. This is the data field from the primary array's first data cell. The COSTSn and SALESn fields are the cost and sales amounts from the primary array's data cells for columns 1 through 8. This maps an entire row of 8 columns from the primary array to a row with a single column in the secondary array.

In the ARRAYAPP example, the EOFMAIN procedure calls a report procedure named COSTRPT2 to format the array information on a different report using the secondary array structure defined in COSTARRX. Figure 13-20 shows the report procedure COSTRPT2 REPORT command.

```
COSTRPT2: REPORT,
;
; Report Costs/Sales by Day and Week
; Columnize the Days Across the Page using the Overdefined Array
;
              B.WEEKNO,
              B.COSTS1 B.SALES1,
              B.COSTS2 B.SALES2,
              B.COSTS3 B.SALES3,
              B.COSTS4 B.SALES4,
              B.COSTS5 B.SALES5,
              B.COSTS6 B.SALES6,
              B.COSTS7 B.SALES7,
              B.COSTS8 B.SALES8,
        TYPE SUBROUTINE
```

Figure 13-20 Array Examples Application ARRAYAPP: COSTRPT2 Report Procedure (Page 1 of 2)

```
FORMAT SUMMARYLABELS SUPPRESS
TITLE 'REPORT OF COSTS AND SALES BY DAY WITHIN WEEK ',
      '- AMOUNTS IN $1000'
ITEM B.WEEKNO SPACES 1
ITEM B.COSTS1 B.SALES1,
     B.COSTS2 B.SALES2,
     B.COSTS3 B.SALES3,
     B.COSTS4 B.SALES4,
     B.COSTS5 B.SALES5,
     B.COSTS6 B.SALES6,
     B.COSTS7 B.SALES7,
     B.COSTS8 B.SALES8,
  PICTURE P'ZZZ9.9'
TOTAL B.COSTS1 B.SALES1,
      B.COSTS2 B.SALES2,
      B.COSTS3 B.SALES3,
      B.COSTS4 B.SALES4,
      B.COSTS5 B.SALES5,
      B.COSTS6 B.SALES6,
      B.COSTS7 B.SALES7,
      B.COSTS8 B.SALES8,
  AT LEVEL G
SECTION COLUMNHEADING
LINE 'WEEK   ....MONDAY...   ...TUESDAY...   ..WEDNESDAY..',
     '   ...THURDAY...   ....FRIDAY...   ...SATURDAY..',
     '   ....SUNDAY...   ..WEEK TOTAL.'
LINE ' NO.   COSTS   SALES   COSTS   SALES   COSTS   SALES',
     '   COSTS   SALES   COSTS   SALES   COSTS   SALES',
     '   COSTS   SALES   COSTS   SALES'
END SECTION
END REPORT
```

Figure 13-20 Array Examples Application ARRAYAPP: COSTRPT2 Report Procedure (Page 2 of 2)

**Note:** For a complete version of this example, see Array Examples Application ARRAYAPP: COSTRPT2 Report Procedure – Part 6 of 10 on page D-103.

The qualifier B refers to the array fields. The B qualifier was assigned to the secondary array in the application on the ARRAY command (see Figure 13-13 on page 13-16). Grand totals (at level G) are produced for each cost and sales amount field on the report. The PICTURE keyword edits the amount fields without commas (ZZZ9.9).

To customize column headings, use the SECTION COLUMNHEADING to "paint" the column headings. This creates a Formatted Section Reporting, Column-Heading section. See Chapter 14, *Extended Reporting Capabilities* for more details.

Figure 13-21 shows the array data formatted by the specifications in the COSTRPT2 report procedure.

11/06/01 REPORT OF COSTS AND SALES BY DAY WITHIN WEEK - AMOUNTS IN $1000 PAGE 1

| WEEK | ....MONDAY... | | ...TUESDAY... | | ..WEDNESDAY.. | | ...THURDAY... | | ....FRIDAY... | | ...SATURDAY.. | | ....SUNDAY... | | ..WEEK TOTAL. | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| NO. | COSTS | SALES | COSTS | SALES | COSTS | SALES | COSTS | SALES | COSTS | SALES | COSTS | SALES | COSTS | SALES | COSTS | SALES |
| 01 | 2.2 | 10.0 | 2.2 | 10.0 | 2.2 | 10.0 | 2.2 | 10.0 | 2.2 | 10.0 | 2.2 | 10.0 | 2.2 | 10.0 | 15.4 | 70.0 |
| 02 | 2.2 | 9.8 | 2.2 | 9.8 | 2.2 | 9.8 | 2.2 | 9.8 | 2.2 | 9.8 | 2.2 | 9.8 | 2.2 | 9.8 | 15.4 | 68.6 |
| 03 | 2.1 | 9.6 | 2.1 | 9.6 | 2.1 | 9.6 | 2.1 | 9.6 | 2.1 | 9.6 | 2.1 | 9.6 | 2.1 | 9.6 | 14.7 | 67.2 |
| 04 | 2.1 | 9.2 | 2.1 | 9.3 | 2.1 | 9.3 | 2.1 | 9.3 | 2.1 | 9.3 | 2.1 | 9.3 | 2.1 | 9.3 | 14.7 | 65.0 |
| 05 | 2.2 | 9.7 | 2.2 | 9.8 | 2.2 | 9.8 | 2.2 | 9.8 | 2.2 | 9.8 | 2.2 | 9.8 | 2.2 | 9.8 | 15.4 | 68.5 |
| 06 | 2.1 | 9.3 | 2.1 | 9.3 | 2.1 | 9.3 | 2.1 | 9.3 | 2.1 | 9.3 | 2.1 | 9.3 | 2.1 | 9.3 | 14.7 | 65.1 |
| 07 | 2.1 | 9.6 | 2.1 | 9.6 | 2.1 | 9.6 | 2.1 | 9.6 | 2.1 | 9.6 | 2.1 | 9.6 | 2.1 | 9.6 | 14.7 | 67.2 |
| 08 | 2.2 | 9.8 | 2.2 | 9.8 | 2.2 | 9.8 | 2.2 | 9.8 | 2.2 | 9.8 | 2.2 | 9.8 | 2.2 | 9.8 | 15.4 | 68.6 |
| 09 | 2.1 | 9.2 | 2.1 | 9.2 | 2.1 | 9.2 | 2.1 | 9.2 | 2.1 | 9.2 | 2.1 | 9.2 | 2.1 | 9.2 | 14.7 | 64.4 |
| 10 | 2.1 | 9.3 | 2.1 | 9.4 | 2.1 | 9.4 | 2.1 | 9.4 | 2.1 | 9.4 | 2.1 | 9.4 | 2.1 | 9.4 | 14.7 | 65.7 |
| 11 | 2.1 | 9.4 | 2.1 | 9.4 | 2.1 | 9.4 | 2.1 | 9.4 | 2.1 | 9.4 | 2.1 | 9.4 | 2.1 | 9.5 | 14.7 | 65.9 |
| 12 | 2.1 | 9.3 | 2.1 | 9.3 | 2.1 | 9.3 | 2.1 | 9.3 | 2.1 | 9.3 | 2.1 | 9.3 | 2.1 | 9.4 | 14.7 | 65.2 |
| 13 | 2.2 | 10.0 | 2.2 | 10.0 | 2.2 | 10.0 | 2.2 | 10.0 | 2.2 | 10.0 | 2.2 | 10.0 | 2.2 | 10.0 | 15.4 | 70.0 |
| 14 | 2.2 | 9.9 | 2.2 | 9.9 | 2.2 | 9.9 | 2.2 | 9.9 | 2.2 | 9.9 | 2.2 | 9.9 | 2.2 | 9.9 | 15.4 | 69.3 |
| 15 | 2.1 | 9.5 | 2.2 | 9.7 | 2.2 | 9.7 | 2.2 | 9.7 | 2.2 | 9.7 | 2.2 | 9.7 | 2.2 | 9.7 | 15.3 | 67.7 |
| 16 | 2.3 | 10.1 | 2.3 | 10.1 | 2.3 | 10.1 | 2.3 | 10.1 | 2.3 | 10.1 | 2.3 | 10.1 | 2.3 | 10.1 | 16.1 | 70.7 |
| 17 | 2.1 | 9.6 | 2.1 | 9.6 | 2.1 | 9.6 | 2.1 | 9.6 | 2.2 | 9.8 | 2.2 | 9.8 | 2.2 | 9.8 | 15.0 | 67.8 |
| 18 | 2.2 | 10.0 | 2.2 | 10.0 | 2.2 | 10.0 | 2.3 | 10.1 | 2.3 | 10.1 | 2.3 | 10.1 | 2.3 | 10.1 | 15.8 | 70.4 |
| 19 | 2.3 | 10.3 | 2.3 | 10.3 | 2.3 | 10.3 | 2.3 | 10.3 | 2.3 | 10.3 | 2.3 | 10.3 | 2.3 | 10.3 | 16.1 | 72.1 |
| 20 | 2.3 | 10.5 | 2.4 | 10.6 | 2.4 | 10.6 | 2.4 | 10.6 | 2.4 | 10.6 | 2.4 | 10.6 | 2.4 | 10.6 | 16.7 | 74.1 |
| 21 | 2.4 | 10.6 | 2.4 | 10.6 | 2.4 | 10.6 | 2.4 | 10.6 | 2.4 | 10.8 | 2.4 | 10.8 | 2.4 | 10.8 | 16.8 | 74.8 |
| 22 | 2.4 | 10.8 | 2.4 | 10.8 | 2.4 | 10.8 | 2.4 | 10.8 | 2.4 | 10.8 | 2.4 | 10.8 | 2.4 | 10.8 | 16.8 | 75.6 |
| 23 | 2.4 | 10.7 | 2.4 | 10.7 | 2.4 | 10.7 | 2.4 | 10.8 | 2.4 | 10.8 | 2.4 | 10.8 | 2.4 | 10.8 | 16.8 | 75.3 |
| 24 | 2.4 | 10.7 | 2.4 | 10.7 | 2.4 | 10.7 | 2.4 | 10.7 | 2.4 | 10.7 | 2.4 | 10.7 | 2.4 | 10.7 | 16.8 | 74.9 |
| 25 | 2.5 | 11.3 | 2.5 | 11.3 | 2.5 | 11.3 | 2.5 | 11.3 | 2.5 | 11.3 | 2.5 | 11.3 | 2.5 | 11.3 | 17.5 | 79.1 |
| 26 | 2.5 | 11.1 | 2.5 | 11.1 | 2.5 | 11.1 | 2.5 | 11.1 | 2.5 | 11.1 | 2.5 | 11.1 | 2.5 | 11.1 | 17.5 | 77.7 |
| 27 | 2.6 | 11.5 | 2.6 | 11.5 | 2.6 | 11.5 | 2.6 | 11.5 | 2.6 | 11.5 | 2.6 | 11.7 | 2.6 | 11.7 | 18.2 | 80.9 |
| 28 | 2.7 | 12.0 | 2.7 | 12.0 | 2.7 | 12.0 | 2.7 | 12.0 | 2.7 | 12.0 | 2.7 | 12.1 | 2.7 | 12.1 | 18.9 | 84.2 |
| 29 | 2.6 | 11.6 | 2.6 | 11.6 | 2.6 | 11.6 | 2.6 | 11.6 | 2.6 | 11.6 | 2.6 | 11.6 | 2.6 | 11.6 | 18.2 | 81.2 |
| 30 | 2.7 | 11.9 | 2.7 | 12.0 | 2.7 | 12.0 | 2.7 | 12.0 | 2.7 | 12.0 | 2.7 | 12.0 | 2.7 | 12.0 | 18.9 | 83.9 |
| 31 | 2.7 | 12.1 | 2.8 | 12.3 | 2.8 | 12.3 | 2.8 | 12.3 | 2.8 | 12.3 | 2.8 | 12.3 | 2.8 | 12.3 | 19.5 | 85.9 |
| 32 | 2.8 | 12.4 | 2.8 | 12.4 | 2.8 | 12.4 | 2.8 | 12.4 | 2.8 | 12.4 | 2.8 | 12.4 | 2.8 | 12.4 | 19.6 | 86.8 |
| 33 | 3.0 | 13.3 | 3.0 | 13.3 | 3.0 | 13.3 | 3.0 | 13.3 | 3.0 | 13.3 | 3.0 | 13.3 | 3.0 | 13.3 | 21.0 | 93.1 |
| 34 | 2.9 | 13.1 | 3.0 | 13.3 | 3.0 | 13.3 | 3.0 | 13.3 | 3.0 | 13.3 | 3.0 | 13.3 | 3.0 | 13.3 | 20.9 | 92.9 |
| 35 | 3.0 | 13.2 | 3.0 | 13.2 | 3.0 | 13.2 | 3.0 | 13.2 | 3.0 | 13.2 | 3.0 | 13.2 | 3.0 | 13.2 | 21.0 | 92.4 |
| 36 | 3.1 | 13.9 | 3.1 | 13.9 | 3.1 | 13.9 | 3.1 | 13.9 | 3.1 | 13.9 | 3.1 | 13.9 | 3.1 | 13.9 | 21.7 | 97.3 |
| 37 | 3.1 | 13.8 | 3.1 | 13.8 | 3.1 | 13.8 | 3.1 | 13.8 | 3.1 | 13.8 | 3.1 | 13.8 | 3.1 | 13.8 | 21.7 | 96.6 |
| 38 | 3.2 | 14.1 | 3.2 | 14.1 | 3.2 | 14.3 | 3.2 | 14.3 | 3.2 | 14.3 | 3.2 | 14.3 | 3.2 | 14.3 | 22.4 | 99.7 |
| 39 | 3.3 | 14.7 | 3.3 | 14.7 | 3.3 | 14.7 | 3.3 | 14.7 | 3.3 | 14.7 | 3.3 | 14.7 | 3.3 | 14.7 | 23.1 | 102.9 |
| 40 | 3.4 | 15.0 | 3.4 | 15.2 | 3.4 | 15.2 | 3.4 | 15.2 | 3.4 | 15.2 | 3.4 | 15.2 | 3.4 | 15.2 | 23.8 | 106.2 |
| 41 | 3.4 | 15.4 | 3.5 | 15.6 | 3.5 | 15.6 | 3.5 | 15.6 | 3.5 | 15.6 | 3.5 | 15.6 | 3.5 | 15.6 | 24.4 | 109.0 |
| 42 | 3.5 | 15.7 | 3.5 | 15.7 | 3.5 | 15.7 | 3.5 | 15.7 | 3.5 | 15.7 | 3.5 | 15.7 | 3.5 | 15.7 | 24.5 | 109.9 |
| 43 | 3.5 | 15.8 | 3.6 | 16.0 | 3.6 | 16.0 | 3.6 | 16.0 | 3.6 | 16.0 | 3.6 | 16.0 | 3.6 | 16.0 | 25.1 | 111.8 |
| 44 | 3.6 | 16.2 | 3.6 | 16.2 | 3.6 | 16.2 | 3.6 | 16.2 | 3.6 | 16.2 | 3.6 | 16.2 | 3.6 | 16.2 | 25.2 | 113.4 |
| 45 | 3.8 | 16.8 | 3.8 | 16.8 | 3.8 | 16.8 | 3.8 | 16.8 | 3.8 | 16.8 | 3.8 | 16.8 | 3.8 | 16.8 | 26.6 | 117.6 |
| 46 | 3.8 | 16.9 | 3.8 | 16.9 | 3.8 | 16.9 | 3.8 | 16.9 | 3.8 | 16.9 | 3.8 | 16.9 | 3.8 | 16.9 | 26.6 | 118.3 |
| 47 | 3.4 | 15.2 | 3.4 | 15.2 | 3.4 | 15.2 | 3.4 | 15.2 | 3.4 | 15.2 | 3.4 | 15.2 | 3.4 | 15.2 | 23.8 | 106.4 |
| 48 | 3.5 | 15.8 | 3.5 | 15.8 | 3.5 | 15.8 | 3.5 | 15.8 | 3.5 | 15.8 | 3.5 | 15.8 | 3.5 | 15.8 | 24.5 | 110.6 |
| 49 | 3.5 | 15.7 | 3.5 | 15.7 | 3.5 | 15.7 | 3.5 | 15.7 | 3.5 | 15.7 | 3.5 | 15.7 | 3.5 | 15.7 | 24.5 | 109.9 |
| 50 | 3.6 | 16.1 | 3.6 | 16.1 | 3.6 | 16.1 | 3.6 | 16.1 | 3.6 | 16.1 | 3.6 | 16.1 | 3.6 | 16.1 | 25.2 | 112.7 |
| 51 | 3.7 | 16.5 | 3.7 | 16.5 | 3.7 | 16.5 | 3.7 | 16.5 | 3.7 | 16.5 | 3.7 | 16.5 | 3.7 | 16.5 | 25.9 | 115.5 |
| 52 | 3.8 | 16.9 | 3.8 | 16.9 | 3.8 | 16.9 | 3.8 | 16.9 | 3.8 | 16.9 | 3.8 | 16.9 | 3.8 | 16.9 | 26.6 | 118.3 |
| | 142.1 | 634.9 | 142.7 | 636.6 | 142.7 | 636.8 | 142.8 | 637.0 | 142.9 | 637.4 | 142.9 | 637.7 | 142.9 | 637.9 | 999.0 | 4458.3 |

Figure 13-21 Report Produced by the COSTRPT2 Report Procedure in the ARRAYAPP Application

Notice that the report looks just like the way the array structure is visualized, columns of days and rows of weeks. The customized column headings along with the totals of the columns adds clarity and completeness to the report.

By using array equivalence to overdefine the primary array with a secondary view of the structure, you were able to exceed the basic application requirements. All you did was map the work area to match the way you wanted to view the structure.

## Output an Array as a Subfile

Once an array record is built and loaded with data and information in memory, you may want to save the entire array record structure for future application processing. This can be easily accomplished by using an output subfile and specifying that a copy of the entire array record be selected for output. See Chapter 7, *Output File Processing* for details on subfile output.

In the ARRAYAPP application, the EOFMAIN procedure calls a subfile (EXTRACT) procedure named ARROUT that outputs the entire COSTARR to a subfile. Figure 13-22 shows the EXTRACT command for the subfile procedure ARROUT.

```
ARROUT: PROC TYPE SUBROUTINE
;
; Output the Entire Internal Array to a Subfile
; ( This will become Last Year's Data at next Year End )
;
 EXTRACT FILE ARRAYOUT ENTIRE A RECFM KEYVSAM
;
END PROC
```

Figure 13-22 Array Examples Application ARRAYAPP: ARROUT Procedure

**Note:** For a complete version of this example, see Array Examples Application ARRAYAPP: ARROUT Procedure – Part 8 of 10 on page D-105.

The Destination Name ARRAYOUT links you to the subfile (M4SUBF1) FILE SUBFI command in the Run Control group (see Figure 13-13). The ENTIRE keyword specifies an A that is the qualifier for the COSTARR array. By specifying a work area qualifier, VISION:Builder knows that you want the entire work area record and structure output to the subfile data set. The RECFM KEYVSAM keyword specifies the output data set for the subfile is a VSAM KSDS format. VISION:Builder performs the appropriate access method I/O automatically. An entire array record can be output just that easily.

Individual data cell fields can be selected for subfile output just like any other work area data field in an application. Also, by using array equivalence to overdefine rows, along with controlled array looping, the output possibilities are unlimited.

## Input an Array as a Coordinated File

Once you have written out an entire array record, you may need to read the entire array record as input to an application. This is done using the coordinated input file mechanism. See Chapter 6, *Input File Processing* for details on coordinated input files.

You include an input coordinated file in your application and indicate that the input record is an array structure. The array definition is used for the input coordinated file definition. When and how the array records are read depends on the type of coordinated file processing you specify. Once they are read into

memory, the array record is available to your application for array processing in the coordinated file work area. The input coordinated file processing mechanism loads the array record structure into memory.

In the ARRAYAPP application, the EOFMAIN procedure calls a subroutine procedure named ARRIN, initiating an input read of an indexed coordinated file that contains an array record. The input coordinated file (M4CORD2) is specified in the FILE CORDn command in the Run Control command group as indexed coordination and is indicated as containing array records. Qualifier 2 is assigned to this external array and the COSTARR array definition is used for the file definition. See the Run Control commands in Figure 13-13. The array record is the COSTARR array from the previous year. Figure 13-23 shows the ARRIN procedure.

```
ARRIN: PROC TYPE SUBROUTINE
;
; Read an Array into memory as Cord File 2 - Indexed
;
LASTYEAR: FIELD C 4                    ; ICF Key Field for Cord 2 Read
;
LET T.LASTYEAR = A.YEAR - 1
;
IF T.LASTYEAR = 2.YEAR THEN            ; <- This causes the ICF Read
;
;  Report Last Year's Array Data
;
   REPORT 2.YEAR 2.WEEKNO 2.DAYNO 2.COSTS 2.SALES
          FORMAT WIDTH 80 DATEFMT TODAYX
          GROUP BY 2.YEAR AT LEVEL 1 SUBTITLE NEWPAGE
          ITEM 2.YEAR SPACES 38
   TITLE '                          COST AND SALES BY DAY AND',
         ' WEEK                             '
   TITLE '                        AMOUNTS SHOWN IN 1000 OF D',
         'OLLARS                            '
   TITLE '                       *******  LAST YEARS DATA  *',
         '*******                           '
   END REPORT
;
ELSE
;
;  There was no data for last year
;
END IF
;
END PROC
```

Figure 13-23 Array Examples Application ARRAYAPP: ARRIN Procedure

**Note:** For a complete version of this example, see Array Examples Application ARRAYAPP: ARRIN Procedure – Part 7 of 10 on page D-104.

This procedure is an example and includes the minimum number of statements. The LET statement is used to put a value into the temporary field T.LASTYEAR. This is the key value of the indexed coordinated file record that you want to read. The IF statement initiates the indexed read by referencing a field from the M4CORD2 file. If an indexed record is found with the key value equal to T.LASTYEAR, you could perform some processing of the input array record. In this example, we produced a simple report.

There is nothing special regarding array input records. They are stored on a file just like any other record. They are read into memory just like any other file using the coordinated input file mechanism. Once in memory, VISION:Builder allows you to process the record structure as an array record structure.

You can read files that contain array records using a standard VISION:Builder file definition. However, the file definition must map the record according to how the array structure was stored. The array record is stored as if it contained two segment types: a root segment consisting of one field, the record key, and one level two segment consisting of the data cell fields. The level two segment occurs a fixed number of times equal to the number of rows times the number of columns defined for the associated array record.

Figure 13-24 shows you a File Definition Glossary Listing that describes to VISION:Builder the array record as a simple two-level structure. Figure 13-25 on page 13-30 shows a simple application that reads the array record as a master input file and outputs the data as a simple report. This is included here just to show you how it works.

```
NOV 05, 2001  16.56.48                                                                                    PAGE     12
                                                FILE STRUCTURE DIAGRAM FOR                                DIAGRAM PAGE  1
                                                FILE DEFINITION - FD4ARRAY
 RECORD FORMAT = KEY SEQUENCED VSAM                                                  FILE IDENTIFICATION =
 RECORD SIZE = 7492    BUFFER SIZE = 7492                                            NUMBER OF SEGMENTS IN FILE = 2

                                                      +---------+
                                                      °SEG001L1 °
                                                      +---------+
                                                           °
                                                           °
                                                           °
                                                      +---------+
                                                      °SEG002L2 °
                                                      +---------+
 NOV 05, 2001  16.56.48                                                                                    PAGE     13
                                         DETAILED GLOSSARY BY LOCATION FOR                                GLOSSARY PAGE      1
                                            FILE DEFINITION - FD4ARRAY
 DATE CATALOGED - 11/05/01 16:56:48                                          DATE LAST UPDATED - 11/05/01 16:56:48
 CATALOGED BY -                                                              LAST UPDATED BY -
 EXPIRATION DATE -                                                           DATE LAST USED -
 FILE IDENTIFICATION =                                                       RECORD FORMAT = KEY SEQUENCED VSAM
 NUMBER OF SEGMENTS IN FILE = 2                                              RECORD SIZE = 7492
 NUMBER OF FIELDS IN FILE = 7                                                BUFFER SIZE = 7492

                                          ********************************
                                          *    SEGMENT   1, LEVEL 1     *
                                          ********************************
 SEGMENT OCCURS N TIMES = 1                                      KEY FIELD 1 = YEAR       TYPE = C   LENGTH = 4
 SEGMENT SIZE = 4
 NUMBER OF FIELDS IN SEGMENT = 1
 ------------------------------------------------------------------------------------------------------------------------
 FIELD   FIELD   FIELD    FIELD   FIELD    DEC    CNT FIELD  EDIT CODES     EDIT    OUTPUT   LINE  DATA BASE NAME/
 NAME    TYPE   LOCATION  LENGTH  RNDING  PLACES  FOR SGMT   ( ) ( ) ( )   LENGTH   WIDTH     NO   *** COLUMN HEADING ***
 ------------------------------------------------------------------------------------------------------------------------
 YEAR      C        1        4                                              4        4
                                                                                          1    ***      YEAR      ***
                                                                                          2    ***      NO.       ***
 NOV 05, 2001  16.56.48                                                                                    PAGE     14
                                         DETAILED GLOSSARY BY LOCATION FOR                                GLOSSARY PAGE      2
                                            FILE DEFINITION - FD4ARRAY

                                          ********************************
                                          *    SEGMENT   2, LEVEL 2     *
                                          ********************************
 SEGMENT OCCURS N TIMES = 416                                    KEY FIELD 1 = KEYSEG2    TYPE = C   LENGTH = 3
 SEGMENT SIZE = 18
 NUMBER OF FIELDS IN SEGMENT = 6
 ------------------------------------------------------------------------------------------------------------------------
 FIELD   FIELD   FIELD    FIELD   FIELD    DEC    CNT FIELD  EDIT CODES     EDIT    OUTPUT   LINE  DATA BASE NAME/
 NAME    TYPE   LOCATION  LENGTH  RNDING  PLACES  FOR SGMT   ( ) ( ) ( )   LENGTH   WIDTH     NO   *** COLUMN HEADING ***
 ------------------------------------------------------------------------------------------------------------------------
 KEYSEG2   C        1        3                                              3        3
 WEEKNO    C        1        2                                              2        4
```

Figure 13-24 File Definition Glossary Listing (Page 1 of 2)

```
                                                                                1   ***    WEEK     ***
                                                                                2   ***    NO.      ***
DAYNO    C       3      1                                      1      3
                                                                                1   ***    DAY      ***
                                                                                2   ***    NO.      ***
HOURS    Z       4      5          1                           8      8
                                                                                1   ***   HOURS     ***
                                                                                2   ***   WORKED    ***
COSTS    Z       9      5          1                           8      8
                                                                                1   ***   COSTS     ***
                                                                                2   ***   1000'S    ***
SALES    Z      14      5          1                           8      8
                                                                                1   ***   SALES     ***
                                                                                2   ***   1000'S    ***
** CLSGB00  TYPE 0  END OF FILE DEFINITION GLOSSARY.                                                  *****
```

Figure 13-24 File Definition Glossary Listing (Page 2 of 2)

```
 CONTROL NAME F366 TERM LISTGEN

 FILE MASTER INPUT FD4ARRAY KEYS NONE
 FILE REPORT
 ;
 ; This job processes the array that was produced from
 ; the Array Application and written out as a Subfile.
 ;
 ; We will process Array as if it was a File and Report the Data.
 ; This is done to show you how it works and help visualize
 ; how the Array rows and cells are structured.
 ;
REPORT O.YEAR O.WEEKNO O.DAYNO O.COSTS O.SALES

   FORMAT WIDTH 80 DATEFMT TODAYX
   GROUP BY O.YEAR
   TITLE 'LIST THE ARRAY THAT WAS OUTPUT'
   TITLE 'AS A SUBFILE - IT IS PROCESSED'
   TITLE 'AS IF IT WERE A TWO LEVEL FILE'
 END REPORT
```

Figure 13-25 Application to Read an Array as a Master Input File

**Note:** For a complete version of this example, see Application To Read An Array As A Master Input File – Part 1 of 2 on page D-120.

Figure 13-25 shows a portion of the Report Output from the application.

```
11/06/2001              LIST THE ARRAY THAT WAS OUTPUT            PAGE 1
                        AS A SUBFILE - IT IS PROCESSED
                        AS IF IT WERE A TWO LEVEL FILE
-------------------------------------------------------------------------------
  YEAR  WEEK  DAY   COSTS     SALES
  NO.   NO.   NO.   1000'S    1000'S
-------------------------------------------------------------------------------
  2002   01    1       2.2      10.0
         01    2       2.2      10.0
         01    3       2.2      10.0
         01    4       2.2      10.0
         01    5       2.2      10.0
         01    6       2.2      10.0
         01    7       2.2      10.0
         01    8      15.4      70.0
         02    1       2.2       9.8
         02    2       2.2       9.8
         02    3       2.2       9.8
         02    4       2.2       9.8
         02    5       2.2       9.8
         02    6       2.2       9.8
         02    7       2.2       9.8
         02    8      15.4      68.6
         03    1       2.1       9.6
                 .
                 .
                 .
```

Figure 13-26 Report Output (Page 1 of 2)

```
          51    1        3.7      16.5
          51    2        3.7      16.5
          51    3        3.7      16.5
          51    4        3.7      16.5
          51    5        3.7      16.5
          51    6        3.7      16.5
          51    7        3.7      16.5
          51    8       25.9     115.5
          52    1        3.8      16.9
          52    2        3.8      16.9
          52    3        3.8      16.9
          52    4        3.8      16.9
          52    5        3.8      16.9
          52    6        3.8      16.9
          52    7        3.8      16.9
          52    8       26.6     118.3
** MK4FT03  TYPE 0  END OF REPORT.                                                                *****
```

Figure 13-26 Report Output (Page 2 of 2)

Chapter

# 14 Extended Reporting Capabilities

Standard reports were introduced in Chapter 9, *Reporting*. The layout of the report page follows a standardized format. The title lines are followed by subtitles, column headings, report detail lines, and summary information. A standard report is output with many format defaults, some of which can be easily adjusted or overridden. Others are completely standardized as to location and content. Standard reports also comply with certain rules. For example, title lines can contain only literals and not data fields.

If you want to customize your report in a way not provided for by the overrides in a standard report, VISION:Builder offers a number of different options. The following sections discuss these options.

In VISION:Workbench for DOS, you can paint a section of a standard report.

With ASL, you can specify a Formatted Sectional report to customize just a section of a standard report or you can specify a Formatted report in which you control the entire report layout. Dynamic Report Line Modification offers another method to enhance the format of a report. The graphics capability in VISION:Builder produces a graphic representation of your report data in addition to your report.

## Formatted Sectional Reporting

Formatted Sectional reporting allows you to selectively format one (or more) sections of a report page while letting VISION:Builder continue to format the rest of the page with standard reporting. Sectional reporting allows you to format the Page Title, Column Heading, and Summary sections.

Sectional reporting is especially useful when reports require data values in the title or column heading areas or when custom labeling and placement of summaries is wanted.

Sections can be used in any combination or omitted as you want. If you omit a section, standard VISION:Builder reporting takes over for that section. For example, if you specify a Page Title section and Summary section, VISION:Builder will use the column headings from the various work area definitions as the default to format the Column Heading section of your report.

In addition to being able to modify your report page, you can also do arithmetic calculations on summary data. This concept introduces another type of field called STEMP. STEMP fields are used to store the results of arithmetic calculations on fields automatically summarized by VISION:Builder and can be reported in the Summary section of your report.

The specifications for Sectional reporting are entered in the SECTION block of the Report Command Group. The Sectional reporting commands override standard VISION:Builder report formatting for each particular section that is specified.

## Formatted Sectional Reporting - Syntax

VISION:Builder reports are divided into multiple sections. These sections are defined in the following table. With standard reporting all of the sections are formatted as described in Chapter 9, *Reporting*.

| Report Section | Definition (Print Lines Included in a Section) |
|---|---|
| Preface | All preface pages. |
| Page Title | All title lines including the date/page line.<br>One blank line following the NEWPAGE Subtitle lines.<br>This section can appear at the bottom of the report page. |
| NEWPAGE Subtitle | All NEWPAGE Subtitle lines.<br>One blank line following the NEWPAGE Subtitle lines. |
| Column Heading | All column heading text and border lines.<br>This section can appear at the bottom of the report page. |
| Subtitle | One blank line preceding the Subtitle lines.<br>All Subtitle lines.<br>One blank line following the Subtitle lines. |
| Detail | All detail lines and all blank lines between detail lines. |
| Summary | One preceding blank line.<br>All summary lines.<br>All blank lines between summary lines.<br>One trailing blank line. |

With Formatted Sectional reporting, you can exercise much greater control over the formatting of the Page Title, Column Heading, and Summary sections. Any sections that you do not want to control continues to be formatted as a standard report.

The position of the Page Title and Column Heading sections (top or bottom of the report page) applies, regardless of whether or not you are formatting that section. You always have the option of positioning the titles or column headings at the top or bottom of the report page when using Formatted Sectional reports.

- The Page Title section contains the page number, date, and title information.
- The Column Heading section contains the report column headings.
- The Summary section contains the formatting of the summaries, including arithmetic computations of the summary data.

If a section is omitted and there is no VISION:Builder default for it (for example, neither Page Title section or TITLE commands in the run), no space will be left for the section on the report page.

The following table lists the commands relevant to each section block in a Formatted Sectional Report. Refere to the *VISION:Builder ASL Reference Guide* for details on the syntax for each command.

| Section | Commands |
|---|---|
| Page Title | LINE, SKIP |
| Column Heading | LINE, SKIP |
| Summary | COMPUTE, DATA, LINE, SKIP |

There are some considerations when using Formatted Sectional reports. If a particular section is specified, it takes precedence and related entries on other commands in the Report Command Group may be affected. These are listed in the following table. If a section is not specified, the entries operate as they would in a standard report.

| Section | Command | Conflicting Parameter | Description | Result |
|---|---|---|---|---|
| Page Title | FORMAT | DATEPOS | Date position | Ignored. |
| | FORMAT | PAGEPOS | Page number position | Ignored. |
| | FORMAT | IMAGES n<br>IMGTITLE<br>PHYPAGE | Single-title, multiple images | Ignored. |
| | TITLE | | Report title text | Ignored. |

| Section | Command | Conflicting Parameter | Description | Result |
|---|---|---|---|---|
| Column Heading | FORMAT | HEADINGS | Column heading type | Does not apply. |
| | FORMAT | BORDER | Column heading character | Does not apply. |
| Summary | REPORT | SINGLE SPACE | Vertical spacing | Vertical spacing on summaries is overridden. |
| | REPORT | GRANDSUMS | Autogrand | Ignored. |
| | FORMAT | SUMMARY LABELS | Labels on summary lines | Ignored. |
| | ITEM | NONPRINT SPACES | Non-print | Column space is not allocated and spaces before column are not used for non-print fields that request summaries. |

There are four types of expressions used to output a Formatted Sectional report: Literals, Field Names, Flag Names, and Summary Fields.

*Literals* print on the report exactly as they appear on the LINE or DATA commands. They are normally used as titles, column headings, and summary labels.

*Field Names* can be specified for output on LINE or DATA commands from any work area in the run or temporary fields. These fields must also be specified in the REPORT command COLUMNS list. If they are not part of the detail line of the report, they must have NONPRINT specified on the ITEM comand.

*Flag Names* can appear on LINE or DATA commands. Flag names other than PAGE, DATE, TODAY, ISDATE, and JULIAN must also appear in the REPORT command COLUMNS list if they are specified on LINE or DATA commands. If they are not part of the detail line of the report, they must have NONPRINT specified on the ITEM comand.

*Summary Fields* appearing on LINE or DATA commands must also have the corresponding summary type specified on the corresponding summary command or a REPORT command COLUMNS operand summary function. They contain the summary value computed by VISION:Builder for a particular summary tape.

## Using Formatted Sectional Reporting

Figure 14-1 shows a report where the formatting of the totals is customized using Summary section specifications. The rest of the report is formatted by standard reporting.

```
JUN 14, 2002          COST AND PRICE VALUE                PAGE 3
                        OF STOCK ON HAND

  PRODUCT GROUP;IN

-----------------------------------------------------------------
   ITEM             ITEM          ESTIMATED    ESTIMATED
  NUMBER            NAME          COST VALUE  PRICE VALUE
-----------------------------------------------------------------
  IN-E13   INFORM USER GUIDE      $4,434.50   $5,825.75
  IN-G61   INFORM TRAINING         1,403.35    2,296.70
  IN-J17   INFORM CONCEPTS         3,252.47    5,322.94
  IN-M41   INFORM MESSAGES         1,351.60    2,063.98
  IN-N19C  INFORM INSTALL CICS       135.41      239.69
  IN-N19I  INFORM INSTALL IMS        467.78      828.02


  PRODUCT GROUP;IN     ESTIMATED            COST VALUE
      TOTALS           COST VALUE           DIFFERENCE
                       ----------           ----------
                       $11,045.11              $910.36

                       ESTIMATED            PRICE VALUE
                      PRICE VALUE           DIFFERENCE
                      -----------           -----------
                       $16,577.08            $2,160.38
```

Figure 14-1 Formatted Sectional Report: Summary Section

For Formatted Sectional reporting, the total number of output lines per page is calculated the same way as standard VISION:Builder reports. The detail prints until reaching a specified control break level. This can mean more than one page of detail. When reaching the control break level, the Summary section (if specified) immediately follows without leaving the customary blank line. If space is wanted, you must include a SKIP command as the first element in the Summary section.

Figure 14-2 shows the entire Report Command Group necessary for this report. The automatic grand summary specification is ignored when you format the Summary section, so if they are desired, specific DATA and LINE commands must be included specifying LEVEL G.

```
VALUE: PROC
LET T.NEWCOST  = 1.09 * UNITCOST
LET T.COSTVALU = T.NEWCOST * QANONHND
LET T.COSTVDIF = QANONHND * (T.NEWCOST - UNITCOST)
LET T.NEWPRICE = 1.15 * ITMPRICE
LET T.SALEVALU = T.NEWPRICE * QANONHND
LET T.SALEVDIF = QANONHND * (T.NEWPRICE - ITMPRICE)
LET T.GROUP = PF(ITEMNO,1,2) WITH EDIT P'PRODUCT GROUP; ##'
;
REPORT  T.GROUP, ITEMNO, ITEMNAME, T.COSTVALU, T.SALEVALU,
        T.COSTVDIF, T.SALEVDIF
   ITEM T.COSTVDIF NOPRINT
   ITEM T.SALEVDIF NOPRINT
     FORMAT DATEFMT DATE
     TITLE 'COST AND PRICE VALUE'
     TITLE 'OF STOCK ON HAND'
     ORDER BY T.GROUP ITEMNO ITEMNAME
     GROUP BY T.GROUP SUBTITLE NEWPAGE
   TOTAL T.COSTVALU BY T.GROUP
   TOTAL T.SALEVALU BY T.GROUP
   TOTAL T.COSTVDIF BY T.GROUP
   TOTAL T.SALEVDIF BY T.GROUP
     SECTION SUMMARY
       SKIP 2 LINES AT LEVEL 1
       LINE T.GROUP SPACES(4) 'ESTIMATED' SPACES(12) 'COST VALUE'
       LINE '      TOTALS' SPACES(11) 'COST VALUE' ,
            SPACES(11) 'DIFFERENCE'
       LINE COL(24) '----------' SPACES(11) '----------'
       DATA COL(22)   TOTAL(T.COSTVALU) AT LEVEL 1
       DATA SPACES(8) TOTAL(T.COSTVDIF) AT LEVEL 1
       LINE                    ; output 2 DATA lines
       SKIP 1 LINE AT LEVEL 1
       LINE COL(24) 'ESTIMATED' SPACES(12) 'PRICE VALUE'
       LINE COL(23) 'PRICE VALUE' SPACES(11) 'DIFFERENCE'
       LINE COL(23) '-----------' SPACES(11) '-----------'
       DATA COL(22)   TOTAL(T.SALEVALU) AT LEVEL 1
       DATA SPACES(8) TOTAL(T.SALEVDIF) AT LEVEL 1
       LINE                    ; output 2 DATA lines
     END SECTION
END REPORT
END PROC
```

Figure 14-2 Formatted Sectional Report Command Group: Summary Section

The source listing for the Summary Section is found in Command Group: Summary Section on page D-127.

The SECTION SUMMARY command identifies this SECTION block. The SKIP command specifies two blank lines prior to printing this section. The "AT LEVEL 1" specifications indicate that everything following on that line prints at a level one control break or more major. At control break levels more minor than one, the entire line is bypassed (that is, no vertical space is taken and no output occurs). The SPACES specifications indicate how many columns are left blank prior to printing what follows. Literals are enclosed in single quotes. Field names are preceded by a qualifier and a period; a blank qualifier defaults to N.

Summary entries can occur in any order with respect to control break level. Summaries at a given level can be preceded or followed on the report by summaries at any other level. Because the Summary section is customized, VISION:Builder does not reserve the first 14 spaces on the detail line for summary labels.

If a command is too long to be specified on one line, terminate the line with a comma, then more lines can be used for continuation. The second LINE command is an example of this.

Output lines must not exceed the width of the page as defined on the FORMAT command. Output line widths are determined by adding the widths of individual detail or summary field information (from work area definitions), widths of literal entries, and spaces between entries as specified by SPACES or COL keywords.

Fields print with their associated edit patterns (either the default pattern or one specified by the user). Field GROUP prints with a user specified edit pattern that precedes the value of the field with the literal 'PRODUCT GROUP:'.

Notice that it appears the same way when printed as a page subtitle. A field prints with the associated number of spaces before column from the ITEM command, which defaults to two. The number of spaces before column for a field that is specified as non-print defaults to zero. This is relevant for positioning fields on the report line. Notice that GROUP is preceded by two spaces.

The next example illustrates a summary only report where both the Page Title and Summary sections are customized. The report is shown in Figure 14-3. The format requested is significantly different from what VISION:Builder produces automatically.

```
DATE: JUN 14, 2002                                                  PAGE: 3

                              SUMMARY OF SALES PER EMPLOYEE


      SALES PERSON: JJK



CUSTOMER NUMBER: 00002
CUSTOMER NAME:   FINANCIAL SERVICES              TOTAL ORDER AMOUNT:          $93

CUSTOMER NUMBER: 00010
CUSTOMER NAME:   HOME IMPROVEMENT SUPPLIES       TOTAL ORDER AMOUNT:         $535

                                          TOTAL SALES PER EMPLOYEE:         $628
```

Figure 14-3 Formatted Sectional Report Summary Only: Page Title and Summary Sections

Figure 14-4 shows the source commands used to produce the report.

```
RUN: PROC
SALEAMT: FIELD P, LEN 5, FLT '$'
  IF ORDERNO EQ ORDERNO  AND  ITEMORD EQ ITEMORD
     LET T.SALEAMT = ITMPRICE * ITMQTYOR
;
  REPORT  ORPERSON CUSTNO CUSTNAME T.SALEAMT  SUMMARYONLY
      FORMAT   DATEFMT DATE   HEADINGS NO
      ORDER BY ORPERSON CUSTNO
      GROUP BY ORPERSON        SUBTITLE  NEWPAGE
      GROUP BY CUSTNO CUSTNAME AT LEVEL 2
      TOTAL T.SALEAMT BY CUSTNO
    ITEM ORPERSON  NOPRINT
    ITEM CUSTNO    NOPRINT
    ITEM CUSTNAME  NOPRINT
    ITEM T.SALEAMT NOPRINT
   ;
   SECTION PAGETITLE
     LINE COL(5) 'DATE: ' F.DATE SPACES(54) 'PAGE: ' F.PAGE
         SKIP 2 LINES
     LINE COL(31) 'SUMMARY OF SALES PER EMPLOYEE'
         SKIP 3 LINES
     LINE COL(7) 'SALES PERSON: ' ORPERSON
         SKIP 4 LINES
   END SECTION
   ;
   SECTION SUMMARY
     LINE 'CUSTOMER NUMBER: '  CUSTNO
     LINE 'CUSTOMER NAME:   '  CUSTNAME,
          '  TOTAL ORDER AMOUNT: '     ,
          TOTAL(T.SALEAMT) AT LEVEL 2
          SKIP 2 LINES
      DATA COL(38) 'TOTAL SALES PER EMPLOYEE: ' ,
           TOTAL(T.SALEAMT) AT LEVEL 1
      LINE AT LEVEL 1    ; line always output, but only
   END SECTION           ; only contains data at level 1
  END REPORT
;
  END IF
END PROC
```

Figure 14-4 Formatted Sectional Report Commands: Summary and Page Title Sections

The source listing for the Summary and Page Title Sections is found in Commands: Summary and Page Title Sections on page D-129.

The procedure selects only valid orders and valid items and calculates the amount of the sale by multiplying the item price by the quantity ordered. The result is placed in the SALEAMT temporary field. This field is totaled for every customer (at level two) and for every salesperson (at level one).

The SUMMARYONLY keyword on the REPORT command requests a summary only report. The SUBTITLE NEWPAGE keywords on the GROUP command forces a skip to a new page for each new salesperson (when the level one control break field ORPERSON changes).

The Page Title section consists of the current date, page number, and the name of the current salesperson.

The Summary section consists of a line with the total sales amount for each customer (at level two). When the salesperson changes (level one), a line prints with the total sales amount for all the customers for that salesperson. After printing the Summary section of the report, VISION:Builder skips to a new page because of

the P subtitle specification. Because VISION:Builder does not skip to a new page until after it has processed a level one break, multiple level two summaries print on the page. This results in all customer information for one salesperson being reported on the same page.

Notice that the SKIP 2 LINES command in the Summary section does not have a level number associated with it, so the two lines are always skipped, even at the most minor break level.

Notice the last LINE command in the Summary section. The line is always output, but contains information only when a level one control break occurs because of the AT LEVEL 1 entry. At control break levels more minor than one (that is, 2-9), the line contains no information, resulting in a blank line being output.

The next example shows the use of the summary keywords and the ability to create custom page titles, column headings, and summary labels. The report shown in Figure 14-5 is a statistical report of inventory and orders for each item.

```
DATE:  JUN 14, 2002                                          PAGE:  1

                    INVENTORY AND ORDERING STATISTICS

   ITEM       ITEM       ORDER    QUANTITY      QUANTITY      QUANTITY
  NUMBER      PRICE      NUMBER   ORDERED     BACK ORDERED     ON HAND
  -------   --------     ------   --------    ------------    --------
  BL-B01       $7.95          -       200               -         249
  BL-E68       27.50      45097       200              90          87
  BL-F43       21.00      10081       200              50          16
  BL-F43       21.00      31565       200              50          16
  BL-H11A      31.00      10987       200              25          56
  BL-H11A      31.00      12568       200              50          56
  BL-H11F      35.00      65012       100              60          50
  BL-T01B     775.00      90311         5               2          15
  BL-T02A   1,000.00          -         5               -          15
  GN-N34       39.95          -        50               -          14
  IN-E13       28.95      11011       200              15         175
  IN-G61       23.50      28422       200              15          85
  IN-G61       23.50      36576       200              50          85
  IN-J17       23.50      39531       200               5         197
  IN-J17       23.50      86071       200              10         197
  IN-M41       28.95      21116       150              10          62
  IN-M41       28.95      31012       150              35          62
  IN-N19C      18.95      53227       150              20          11
  IN-N19C      18.95      89003       150              35          11
  IN-N19I      18.95      53226        50               9          38
  TR-M52       19.95          -        25               -          28
  TR-M55       15.95      70652        25              12           3
  TR-M89       47.50      72568        50              13          19
  TR-M89       47.50      79031        50               5          19
  TR-N94C      39.95          -        25               -          30
  TR-N94I      39.95      21455        25               3          15
  TR-O13        7.95          -        50               -          59
  TR-O17       23.95          -        25               -          25


INVENTORY STATS YTD TO:  MAY 15, 2002
OTAL QUANTITY ORDERED:    3,285     TOTAL BACK ORDERED:             564
MAX ORDERED - ANY ITEM:     200      MIN ORDERED - ANY ITEM:         5
RATIO OF TOTAL QUANTITY ON HAND TO ORDERS:          .515
PERCENTAGE OF TOTAL BACK ORDERED TO TOTAL ORDERS:         17.16%
AVERGE QUANTITY BACK ORDERED:             26
```

Figure 14-5 Formatted Sectional Report: Summary Keywords

The source listing for the Summary Keywords is found in Commands: Summary Keywords on page D-130.

Figure 14-6 shows the source statements used to produce the report.

```
REPORT T.LASTSPDT, ITEMNO, ITMPRICE,
          ORDERNO, ORDQTY, QTYBKORD, QANONHND
   ITEM T.LASTSPDT NOPRINT
   ITEM ITEMNO SPACES 10
   FORMAT DATEFMT DATE
   ORDER BY ITEMNO
   TOTAL   ORDQTY   AT LEVEL G
   TOTAL   QTYBKORD AT LEVEL G
   AVERAGE QTYBKORD AT LEVEL G
   MAX     ORDQTY   AT LEVEL G
   MIN     ORDQTY   AT LEVEL G
   RATIO   QANONHND TO ORDQTY AT LEVEL G
   PERCENT QTYBKORD OF ORDQTY AT LEVEL G
  SECTION PAGETITLE
    SKIP 2 LINES
    LINE 'DATE:  ' F.DATE  SPACES(40) 'PAGE:  ' F.PAGE
    SKIP 1 LINE
    LINE COL(25) 'INVENTORY AND ORDERING STATISTICS'
  END SECTION
  SECTION COLUMNHEADING
    LINE COL(11) 'ITEM           ITEM      ORDER    QUANTITY',
         '      QUANTITY'
    LINE COL(10) 'NUMBER        PRICE      NUMBER   ORDERED' ,
         '    BACK ORDERED'
    LINE COL(10) '------        -----      ------   --------',
         '    ------------'
  END SECTION
  SECTION SUMMARY
    SKIP 5 LINES
    LINE 'INVENTORY STATS YTD TO:  ' T.LASTSPDT    AT LEVEL G
    LINE 'TOTAL QUANTITY ORDERED: '      TOTAL(ORDQTY)  ,
      '   TOTAL QUANTITY BACK ORDERED: ' TOTAL(QTYBKORD),
                                                   AT LEVEL G
    LINE 'RATIO OF TOTAL QUANTITY ON HAND TO ORDERS: '  ,
                                 RATIO(QANONHND)   AT LEVEL G
    LINE 'PERCENTAGE OF TOTAL QUANTITY BACK ORDERED TO ',
                'TOTAL ORDERS: ' PERCENT(QTYBKORD) AT LEVEL G
    LINE 'MAX QUANTITY ORDERED OF ANY ITEM: ' MAX(ORDQTY),
      '   MIN QUANTITY ORDERED OF ANY ITEM: ' MIN(ORDQTY),
                                                   AT LEVEL G
    LINE 'AVERGE QUANTITY BACK ORDERED: ' AVERAGE(QTYBKORD),
                                                   AT LEVEL G
  END SECTION
END REPORT
```

Figure 14-6 Source Statements for Year-to-Date Ordering Statistics

The temporary field LASTSPDT has the NOPRINT keyword specified on the ITEM command so it will not print as part of the detail lines. It prints only in the Summary section. Note that the spacing centers the column headings over the fields in the detail lines. The dashes under the headings make them stand out. Because the Summary section is customized, VISION:Builder does not reserve the first 14 spaces on the detail line for summary labels.

Each report page contains the customized page titles and column headings formatted according to the specifications on the commands within the corresponding section. The detail lines print down the page to fill the height of the

page and VISION:Builder automatically skips to the top of the next page as with standard reporting. Because the entire Summary section is keyed to the grand level, it prints only on the last page of the report.

Figure 14-7 illustrates an invoice written using Formatted Sectional Reporting with a Summary section reporting values that are computed from the VISION:Builder summaries.

```
INVOICE NO: UD25A                                   JUN 14, 2002

       FINANCIAL SERVICES
       CUSTOMER ID:  00002
       (212) 555-1234


                                                              PAGE: 1

**********************************************************************
** ORDER      ITEM     QUANTITY    ITEM         TOTAL               **
** NUMBER    NUMBER     ORDERED    PRICE         COST               **
**********************************************************************
   20103     BL-H11A         3    $31.00    $    93.00
   20105     BL-H11B         3    $35.00    $   105.00


                                        SUBTOTAL       $   198.00
                                        SALES TAX      $    14.36
                                        FREIGHT CHARGE $    11.19
                                                        ---------
                                        INVOICE TOTAL  $   223.55
                                                        =========
```

Figure 14-7 Formatted Sectional Report: Computed Summaries

The source listing for the Computed Summaries is found in Commands: Computed Summaries on page D-132.

Figure 14-8 illustrates the entries required to produce the report.

```
COMPUTE: PROC
   IF ORDERNO = ORDERNO    ; only report customers with orders
         LET T.TOTITMPC = QTYSHIP * ITMPRICE
         LET T.FREIGHT  = FRTCOST + T.FREIGHT
;
REPORT CUSTNAME CUSTNO CUSTPH INVNO T.FREIGHT  ,
       ORDERNO ITEMORD ITMQTYOR ITMPRICE T.TOTITMPC
  FORMAT   DATEFMT DATE
  ORDER BY CUSTNAME, INVNO, ORDERNO
  GROUP BY CUSTNAME        SUBTITLE  NEWPAGE
  GROUP BY CUSTNO, CUSTPH, INVNO, ORDERNO  AT LEVEL 2
  TOTAL T.FREIGHT  T.TOTITMPC  BY CUSTNAME
  ITEM CUSTNAME, CUSTNO, INVNO  NOPRINT
  ITEM CUSTPH     NOPRINT  PICTURE P'(###) ###-####'
  ITEM T.FREIGHT  NOPRINT  PICTURE P'$ZZZZZ9.99'
  ITEM ORDERNO  SPACES 3
  ITEM ITMQTYOR SPACES 0
  ITEM ITEMORD  SPACES 5
  ITEM ITMPRICE SPACES 1    PICTURE P'$$$$$9.99'
  ITEM T.TOTITMPC SPACES 4  PICTURE P'$ZZZZZ9.99'
 SECTION PAGETITLE
```

Figure 14-8 Formatted Sectional Report Commands: Computed Summaries (Page 1 of 2)

```
        SKIP 2 LINES
    LINE 'INVOICE NO: ' INVNO   COL(50) F.DATE
        SKIP 1 LINES
    LINE COL(8) CUSTNAME
    LINE COL(8) 'CUSTOMER ID:  ' CUSTNO
    LINE COL(8) CUSTPH
        SKIP 2 LINES
    LINE COL(63) 'PAGE: ' F.PAGE
        SKIP 2 LINES
END SECTION
SECTION COLUMNHEADING
    LINE LIT('*',70)
    LINE '** ORDER ' SPACES(4) ' ITEM ' SPACES(4) 'QUANTITY' ,
         SPACES(4) 'ITEM ' SPACES(8) 'TOTAL' COL(69) '**'
    LINE '** NUMBER' SPACES(4) 'NUMBER' SPACES(4) ' ORDERED' ,
         SPACES(4) 'PRICE' SPACES(8) ' COST'  COL(69) '**'
    LINE LIT('*',70)
END SECTION
SECTION SUMMARY
  COMPUTE STEMP01 = TOTAL(T.TOTITMPC) * .0725  ,
          ROUNDED, DEC 2, AT LEVEL 1, PIC P'$ZZZZZ9.99'
  COMPUTE STEMP02 = TOTAL(T.TOTITMPC) + STEMP01,
          ROUNDED, DEC 2, AT LEVEL 1
  COMPUTE STEMP03 = TOTAL(T.FREIGHT)  + STEMP02,
          ROUNDED, DEC 2, AT LEVEL 1, PIC P'$ZZZZZ9.99'
  SKIP 3 LINES AT LEVEL 1
  LINE COL(40) 'SUBTOTAL   ' TOTAL(T.TOTITMPC) AT LEVEL 1
  LINE COL(40) 'SALES TAX      ' STEMP01          AT LEVEL 1
  LINE COL(40) 'FREIGHT CHARGE ' TOTAL(T.FREIGHT)  AT LEVEL 1
  LINE COL(56) LIT('-',9)                         AT LEVEL 1
  LINE COL(40) 'INVOICE TOTAL  ' STEMP03          AT LEVEL 1
  LINE COL(56) LIT('=',9)                         AT LEVEL 1
END SECTION
END REPORT
END ; terminate IF logic
END PROC
```

Figure 14-8 Formatted Sectional Report Commands: Computed Summaries (Page 2 of 2)

The Summary section instructs VISION:Builder to perform arithmetic calculations for the invoice totals based on summary values that VISION:Builder computed. To calculate the totals, three STEMP fields are used.

| | |
|---|---|
| STEMP01 | is a result of multiplying the total cost of all items at a level one control break (TOTAL T.TOTITMPC) by the tax rate .0725, giving the sales tax amount for the invoice. |
| STEMP02 | is the result of adding the total cost of all items at a level one control break to the previously computed sales tax amount in the STEMP01 field. This is the first step in computing the invoice grand total. |
| STEMP03 | is the result of adding the FREIGHT charges for the customer to the STEMP02 result. This is the final step in computing the invoice grand total. |

The output edit patterns create more than six digits plus two decimal places for STEMP01 and seven digits plus two decimal places for STEMP03, overriding the VISION:Builder default length for a STEMP field of 15 digits plus a sign.

# Formatted Reporting

Where standard reporting automatically formats the whole report page and Formatted Sectional Reporting allows custom formatting of parts of the page while automatically formatting the rest of the page, Formatted Reporting requires that the entire report page be custom formatted. VISION:Builder automatically edits fields, sorts the report, calculate summaries, and manages printer carriage control as with standard and formatted sectional reporting.

The capabilities unique to formatted reporting make it ideal for many types of preprinted forms, such as invoices.

## Formatted Reporting - Syntax

As stated in Formatted Sectional Reporting - Syntax on page 14-2, you can use the following types of expressions to output a Formatted Sectional Report or a Formatted Report:

- *Literals* — *Literals* print on the report exactly as they appear on the DATA or LINE command. They are normally used as titles, column headings, and summary labels.
- *Field Names* — *Field Names* can be specified for output on a DATA or LINE command from temporary fields or any work area in the run. These fields must also be specified on the REPORT command. If they are not part of the detail line of the report, they must have NOPRINT specified on an ITEM command.

- *Flag Names* — *Flag Names* can appear on a DATA or LINE command. Flag names other than PAGE, DATE, TODAY, ISDATE, and JULIAN must also appear on the REPORT command if they are specified on a DATA or LINE command. If they are not part of the detail line of the report, they must have NOPRINT specified on an ITEM command.
- *Summary Fields* — *Summary Fields* appearing on DATA or LINE commands must also have the corresponding summary type specified for the field in the Report Command Group. They contain the summary value computed by VISION:Builder for a particular summary type.

The fixed format syntax statements contain the formatting specifications in the text portion (positions [14-72]) of the statement. The Fn statements of a formatted report are divided into three groups: the header group, the detail group, and the trailer group. Each group corresponds to an area of the report page.

The report page consists of a header area, a detail area (containing a specific number of detail lines formatted by VISION:Builder), and a trailer area. The order in which these appear on the report page and the number of times they appear varies with the control break level and the type of report (detail report or summary only report). The layout of the page is described in more detail in Using Formatted Reporting on page 14-16.

The following table shows the Sections in the Report Command Group that relate to each area and the information contained in each Section.

| Report Area | SECTION | Entries | |
|---|---|---|---|
| Header | PAGE TITLE<br>COLUMN HEADING | Literals<br>Field names<br>Flag Names | |
| Detail | DETAIL | SIZE, NEWPAGE | |
| Trailer | SUMMARY | Literals<br>Field names<br>Flag Names | Summary Fields<br>COMPUTE statements<br>Level indicators<br>NEWPAGE command |

Formatted Reports are assumed whenever a SECTION DETAIL command is present inside of a SUMMARY section. In Formatted Reports, the layout of the report lines composing the header area and the trailer area is controlled by the use of the DATA, LINE, and SKIP commands within the specified section. The SECTION command specifies whether the layout commands inside the section apply to the header area or the trailer area.

In a Formatted Report, all three areas must be specified and must appear in the order shown here. However, the header or trailer areas can be empty (that is, contain no lines of text).

The Header area contains literals and field values for the report header area of the report page. It typically consists of title lines and column headings (if needed) that contain literals, field names, and flag names. The SECTION PAGETITLE or SECTION COLUMNHEADING commands specify that the layout commands apply to the header area. You can use either section designation to specify the layout of the header area. If both are specified, they are merged together in the order they are specified.

The Detail area consists of only a SIZE command and (optionally) a NEWPAGE command, indicating the number of lines of detail information to print on the report. The detail lines on the report are automatically formatted by VISION:Builder. You specify only the number of lines to be printed. When the report contains detail lines, a SECTION DETAIL command must be present and a SIZE command must be included to specify the number of lines to be reserved for the detail section. In this case, the summary section, if any, will always appear at a fixed position following the space reserved for the detail lines.

**Note:** In Standard Reports and Formatted Sectional Reports, SUMMARY sections will always float so that they appear immediately following the previous section. The layout of the detail lines in a Formatted Report is determined by the order of the COLUMN keyword operands on the REPORT command and the ITEM command controls. When a Formatted Report is a summary-only report, the SECTION DETAIL command may not appear in the report, but a NEWPAGE command must appear inside the SUMMARY section.

The Trailer area contains literals, flag values, field values, summary fields computed by VISION:Builder (for example, TOT N.SALARY), and STEMP fields (calculated in a COMPUTE statement). These elements appear in the report trailer area of the report page but are printed only at the control break level you specify. The SECTION SUMMARY command specifies that the layout commands apply to the trailer group. A significant distinction for Formatted Reports is whether the report contains detail lines or whether the report is a summary-only report.

A summary field must have control break level associated with it when you specify it. For example, TOT N.SALARY AT LEVEL 1 is associated with control break level 1 and prints whenever the current control break level is 1 or more major.

All other elements are associated with the control break level in effect for that report line. You control the control break level that is in effect by specifying a level on the LINE or DATA command. If no control break level is in effect for a given element, the element always prints on the report line. Otherwise, the element prints only when the current control break level is equal to or more major than the one associated with the element.

Vertical and horizontal spacing of report output lines in the header and trailer groups is specified with SKIP commands. Detail line spacing is controlled in the standard manner using the entries on the ITEM command.

## Using Formatted Reporting

The total number of output lines per page is calculated by adding the number of lines required to print the header, the maximum number of detail lines (which is zero for a summary only report), the number of lines required to print the trailer, plus a one line margin. The maximum number of detail lines is computed by multiplying the number of detail lines on the SIZE command by the DETAILSPACING entry on the FORMAT command. For example, SIZE 20 combined with DETAILSPACING 2 for a double spaced report would allocate 40 lines for the detail area.

### Detail Report

Detail lines print until one of the following events occurs:

- *The maximum number of lines specified on the SIZE command prints* – Detail line printing ceases for the page and only the literal text of the trailer area are printed.
- *A control break occurs at the level specified by NEWPAGE or more major* – If the control break is at the level specified by NEWPAGE or more major (numerically lower), detail printing ceases for the page and the trailer area is printed along with any requested summaries.

  VISION:Builder skips the reserved number of lines specified on the SIZE command until it reaches the same relative position on the page as though the maximum number of detail lines had printed.
- *A control break occurs at a level more minor than specified by NEWPAGE* – If the control break is at a level more minor than specified by NEWPAGE (numerically higher), no skipping takes place and one of the following two actions occurs:
  - If there are summaries, detail printing ceases for the page, but the trailer area prints.
  - If there are no summaries, detail printing continues. Control break fields print only if their values have changed.

### Summary Only Reports

The NEWPAGE specification in the Summary Section controls the number of trailer areas printed per page on a summary only report. The header area prints at the top of each page and the trailer areas print on the same page until a control break occurs at level specified by NEWPAGE or more major.

A page break occurs after the trailer area prints at that level. If the page is filled before a control break occurs at the level specified by NEWPAGE, a new page is automatically produced with a header area at the top. If NEWPAGE is not specified, a page break occurs only when the page is filled. A page is considered filled if there is not enough space left to print another full trailer area.

If NEWPAGE is not specified, the default is one summary per page.

Figure 14-9 illustrates the use of Formatted Reporting with preprinted forms.

The form is an invoice with many headings already typeset. Formatted Reporting allows the information to be printed exactly where dictated by the form.

```
                    YOUR COMPANY LETTERHEAD
                     YOUR COMPANY ADDRESS

PREFERRED CUSTOMER:                                 JUN 14, 2002
   HOME IMPROVEMENT SUPPLIES                      CUST ID:  00010
   (343) 555-3434


ORDER #         QTY  ITEM     UNIT PRICE     TOTAL COST
 01101           1  BL-T01B    $475.00          $475

 01104           1  BL-T02B     535.00           535


                                 TOTAL         $1,010
```

Figure 14-9 Use of Formatted Reporting with Preprinted Forms

Figure 14-10 shows the source statements for the report.

```
   REPORT  CUSTNAME CUSTNO CUSTPH ORDERNO ,
           ITMQTYOR ITEMORD ITMPRICE T.SALEAMT
       GROUP BY CUSTNAME CUSTNO CUSTPH AT LEVEL 1
       ORDER BY CUSTNAME
       TOTAL T.SALEAMT AT LEVEL 1
       FORMAT  DETAILSPACING 2 WIDTH 65
       ITEM CUSTNAME CUSTNO  NOPRINT
       ITEM CUSTPH  NOPRINT  PICTURE P'(###) ###-####'
;
SECTION PAGETITLE
       SKIP 6 LINES
      LINE COL(50) F.DATE
      LINE COL(5) CUSTNAME  SPACES(22) CUSTNO
      LINE COL(5) CUSTPH
       SKIP 3 LINES
END SECTION
;
SECTION DETAIL
       SIZE 10 LINES
       NEWPAGE AT LEVEL 1
END SECTION
;
SECTION SUMMARY
      SKIP 1 LINES
      LINE COL(35) 'TOTAL' TOTAL(T.SALEAMT) AT LEVEL 1
END SECTION
;
 END REPORT
```

Figure 14-10 Commands to Generate Report with Preprinted Forms

The source listing for the commands to generate a report with preprinted forms is found in Commands to Generate Report with Preprinted Forms on page D-134.

The procedure selects only the customers who placed an order and calculates the total cost for each item ordered by multiplying the quantity ordered by the unit cost per item.

The FORMAT command specifies double-spacing on the detail lines and restricts the page width to 65 characters.

The ITEM command contains the fields to print only in the header area, so they are specified as NOPRINT. CUSTNAME is the primary sort field and there is a level one control break on each of the fields in the header area. If any of them change, VISION:Builder skips to a new page.

The REPORT command lists the fields to print on the detail lines. T,SALEAMT is the field whose total is computed by VISION:Builder and that prints in the trailer area.

The PAGETITLE section specifies the header area that prints at the top of every page. Six lines are skipped to allow space for the preprinted header. The first LINE command contains the flag DATE (that did not have to be entered on the REPORT command).

The DETAIL section with a SIZE command specifies a maximum of 10 detail lines. However, because the FORMAT command specified double-spacing, VISION:Builder reserves 20 lines for the detail area. The NEWPAGE command specifies that a page break occurs at level one. Because all three fields in the header area have a level one control break, VISION:Builder skips to a new page if any of them change.

The SUMMARY section makes up the trailer area. The summary level is one so the total on SALEAMT prints when a level one control break occurs.

## Variable Length Fields

While variable length fields can be used in Formatted Reporting and Formatted Sectional Reporting, certain considerations must be kept in mind.

In a Formatted Report, variable length fields in the detail area are processed as in a normal VISION:Builder report. However, when preparing to output a folded variable length field, if VISION:Builder calculates that a certain occurrence of the field will not fit on the current page, it will start a new page with that line. This change of page can cause unwanted blank space to appear on each report page. Because with Formatted Reporting you must specify (using the SIZE command) the exact number of detail lines that you want to appear on each report page, the number of lines produced when the variable length fields fold must be taken into account in the structuring of the page.

Variable length fields have one limitation when used in the header and trailer areas of a Formatted Report and in the Page Title, Column Heading, and Summary sections of a Formatted Sectional report. They do not fold. They are handled as character fields where the amount of space on the print line equals the maximum length or output edit length if specified for that field. The output edit length can be used for partial fielding of the data. Unused space on the print line is filled with trailing blanks.

For more information about variable field definitions, see Chapter 4, *Field Characteristics and Handling*.

## When to Use Formatted Sectional Reporting

There are some types of reports that can be cumbersome or cannot be created with Formatted Reporting.

For example, if you want a report with detail lines, summaries, and more detail lines on the same page, you cannot use Formatted Reporting because it only allows one detail area per page and one summary area that prints after the detail lines. However, Formatted Sectional Reporting does allow more than one detail area per page.

In general, unless VISION:Builder automatic summaries must print in an exact location on the report page, use standard or Formatted Sectional Reporting instead of Formatted Reporting.

## Dynamic Report Line Modification

Another VISION:Builder reporting capability, Dynamic Report Line Modification, enhances the layout of a report. You can do additional editing on the print line image before it is output to the actual report. The emphasis is on modifying the individual lines in a report, as opposed to the entire report page or sections of the page as described previously.

In general, use Dynamic Report Line Modification when standard and Formatted Sectional Reporting do not provide the needed output.

During report preparation, dynamic report line modification can:

- Suppress printing of unwanted blank lines.
- Suppress printing of zero value fields.
- Insert blank lines before or after specified printing.
- Insert a zero before the decimal point for values greater than zero and less than one.
- Rearrange data fields.
- Generate ASA control characters on specified lines to allow skipping to selected printer control channels. This includes the capability of not skipping, printing more than one report line on the same physical print line, and mixing fonts on one print line (if printer capability exists).

During calculation, dynamic report line modification can:

- Calculate and output multiple percentages and ratios on one line.
- Provide more or fewer decimal places in output lines than provided by standard reporting.
- Round (half-adjust) numeric output fields.

## Processing

Dynamic report line modification specifications are keyword-oriented and are entered on an XREP command.

During the report phase, the report line is dynamically changed after the line has been formatted by VISION:Builder and before it is output to the M4LIST output device. Standard reports and reports using Dynamic Report Line Modification can be intermixed in the same processing run.

The following table gives a brief description of each keyword presented in the order acted upon by VISION:Builder. The keywords must be spelled precisely as shown in the following table.

| Keyword | Function |
|---|---|
| TITLE LINENO | Exempts a specified number of lines at the top of each page from most of the other keywords. |
| SKIP | Allows the user to specify ASA carriage control characters to be placed on specific output lines. |
| NOSPACELINES | Suppresses output of completely blank lines. |
| NOCALC | Exempts lines meeting specific logical conditions from some dynamic report line calculations and formatting. |
| PERCENT, RATIO, ROUND | Specifies arithmetic operations, allowing division or round based on data in the output line. They are executed in order of input but at this point (after NOCALC and before DECIMALEDIT keyword) in the processing cycle (see Figure 14-11). |
| DECIMALEDIT | Simultaneously suppresses printing of zero value numeric fields and inserts a leading zero before the decimal point for values between zero and one. |
| SUPPRESS SPACE | Suppresses lines having blanks in specified positions of the print line. |

| Keyword | Function <$paranum |
|---|---|
| SPACEBEFORE | Causes a blank to be output before a specified line. |
| SPACEAFTER | Causes a blank to be output after a specified line. |
| LINE SIZE [,MOVEPAGENO] | Reduces the page width on the output page to the specified size. This operation is performed last and overrides all other specifications. The optional parameter (shown in brackets) causes the standard page number to be moved to coincide with the new page width. |

Some keywords require positional indicators with or without edit pictures. These keywords describe the format of the required data and where to find it on the output line. Positional indicators specify the entire field or its starting position and take the form POSnnn or nnn, where nnn signifies a three digit number between 001 and 132. The nnn form is usually combined with a picture (such as 9,999.99) of the field as it is input to and output from dynamic report line modification. In these specifications, the picture is represented by pppp.

Some keywords can use explicit operators that are logical operators. For example, the equal sign (=) is used in its usual manner to indicate that the operation is to take place if the condition is logically true. The asterisk (*) is used as a not equal operator to provide a single character representation that can be printed.

Some keywords can use several types of operands. Refer to the *VISION:Builder ASL Reference Guide* for descriptions of the other operands and their keyword.

Dynamic Report Line Modification does not interfere with normal use of the preface. Preface lines should follow all dynamic report lines. A carriage control character for page skip is output on the first actual print line rather than on a blank line as in standard VISION:Builder processing. This occurs even on the report identification message, because one of the aims of this capability is to speed up output whenever possible.

Dynamic Report Line Modification specifications can be entered in any order, but they are executed in the order shown in the flowchart in Figure 14-11. If multiple entries are valid for keywords or specifications are equal in the hierarchy (for example, PERCENT, RATIO, ROUND), they are executed in order of input. It is possible and often beneficial to use the same print position for both the input data and the result location or to apply an operation to the result of a previous one on the same line (for example, ROUND the result of a RATIO calculation).

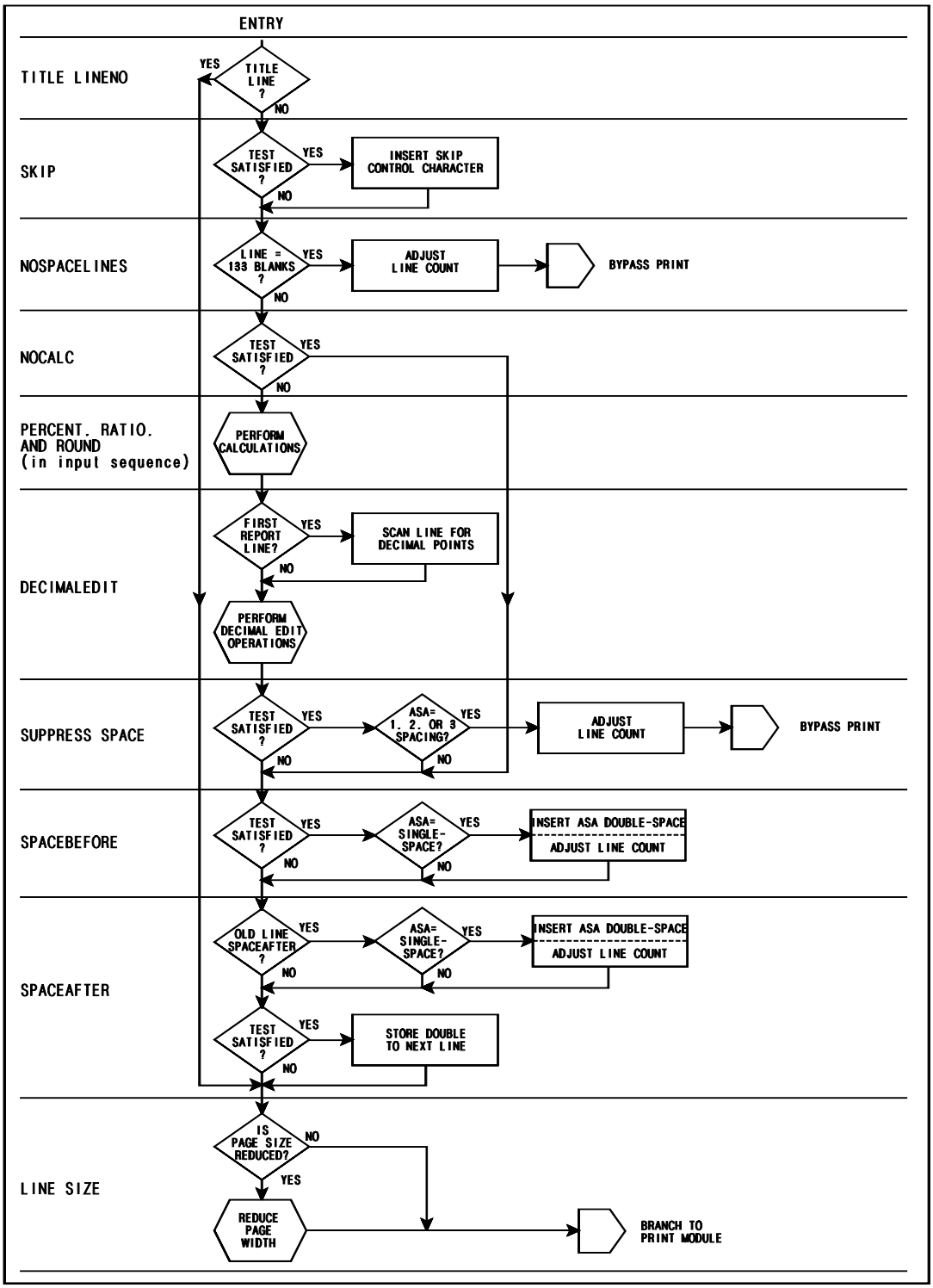



Figure 14-11 Dynamic Report Line Modification Processing

## Errors

When VISION:Builder finds a syntax error in a line that it recognizes as a Dynamic Report Line Modification specification, it outputs the line and a message during the report step of VISION:Builder. Dynamic report line modification continues to scan the rest of the XREP commands in an attempt to locate more errors that are output as found. VISION:Builder outputs the report as if no Dynamic Report Line Modification commands were specified. This allows you to check the output lines being used as input data to Dynamic Report Line Modification.

## Syntax

Dynamic Report Line specifications are entered on the VISION:Builder ASL XREP command or in VISION:Workbench for ISPF.

## Storage Considerations

VISION:Builder does not require extra storage when Dynamic Report Line Modification is used. However, there are some considerations to keep in mind based on the number of operations performed. The following formula should be used in calculating the requirements for the PERCENT, RATIO, ROUND, and DECIMALEDIT keywords. No other keywords have any effect on storage needs.

N = (P*96) + (R*74) + (D*26)

Where: P = number of PERCENT and/or RATIO keywords used in the report.

R = number of ROUND keywords.

D = number of decimal points found in the first line processed by DECIMALEDIT keyword.

The value of N must not exceed 1,270 or Dynamic Report Line Modification will issue an error message and the report is output in standard VISION:Builder format.

## Restrictions and Operating Characteristics

Dynamic report line modification has some operating characteristics that should be noted.

- It is designed to work on standard one-up reports, because it uses the physical output line for its commands and data. Because of this, extra blank preface pages are needed for multiple image reports.
- It executes its keywords in the specific order described in the table starting on page 14-20.
- The DECIMALEDIT keyword is controlled by the location of the decimal points on the first line of data not included in the title lines specified by TITLE LINENO or excluded by NOCALC keyword.
- Folded lines can be difficult to handle.
- No separating characters are allowed on PERCENT and RATIO result fields.
- Positional parameters must have three digits with leading zeros.
- Each specification must be contained on one print line.
- The maximum page width is 132 characters.
- Invalid results of Dynamic Report Line calculations are output as a blank, not an asterisk.

## Using Dynamic Report Line Modification

Figure 14-12 shows the required output. This example produces a summary only report of the sale for each two departments' gross sales and a ratio comparing the two departments. The advantage to Dynamic Report Line Modification is the calculated new column created after all summaries are complete.

```
06/14/02        COMPARISION OF DEPARTMENT GROSS SALES     PAGE 1

-------------------------------------------------------------------
                DEPT   GROSS_A    GROSS_B          RATIO
-------------------------------------------------------------------
GRAND    TOTAL        $1,545.00    $220.50        7.0
```

Figure 14-12 Example of Dynamic Report Line Modification Used to Post-Process a Summary Report

Figure 14-13 shows the same report.

```
06/14/02        COMPARISION OF DEPARTMENT GROSS SALES     PAGE 1

-------------------------------------------------------------------
                DEPT   GROSS_A    GROSS_B          RATIO
-------------------------------------------------------------------
GRAND    TOTAL        $1,545.00    $220.50
```

Figure 14-13 Report Before Customization with Dynamic Report Line Modification

To achieve the finished product, you need to know the edit patterns and start locations of some of the fields. If you run a sample report only run CONTROL command with the SAMPLE keyword, you get the maximum value of each of the numeric fields positioned on the page in Figure 14-14. You can measure the line to get the start positions and the edit pattern as shown.

```
06/14/02        COMPARISION OF DEPARTMENT GROSS SALES     PAGE 1

-------------------------------------------------------------------
                DEPT   GROSS_A    GROSS_B          RATIO
-------------------------------------------------------------------
GRAND    TOTAL       $10,000.00 $10,000.00
```

Figure 14-14 Sample Report Before Customization with Dynamic Report Line Modification

Now you can write the statements to calculate a ratio in the last column, and round the result.

Figure 14-15 shows the dynamic report line source statements for the report in Figure 14-12.

```
REPORT  T.DEPT T.GROSS_A T.GROSS_B T.RATIO  SUMMARYONLY
   FORMAT WIDTH 70
   ORDER BY T.DEPT
   ORDER BY T.DEPT
   GROUP BY T.DEPT
   TOTAL T.GROSS_A T.GROSS_B AT LEVEL G
   TITLE 'COMPARISION OF DEPARTMENT GROSS SALES'
   XREP 'TITLE LINENO=5'
   XREP 'RATIO A=(024,9,999),B=(037,999),C=(050,99.99)'
   XREP 'ROUND A=(050,99.99)'
END REPORT
```

Figure 14-15 Commands to Generate Report Using Dynamic Report Line Modification

The source listing for the commands to generate a report using Dynamic Report Line Modification is found in Commands to Generate Report Using Dynamic Report Line Modification on page D-135.

## Dynamic Report Line Modification with Multi-Image Reports

Multi-image reports such as mailing labels can be processed by Dynamic Report Line Modification. In fact, Dynamic Report Line Modification makes no distinction between multi-image and standard (single image) reports. This is because it always views a report line as a full 132 characters wide.

Consider a page of VISION:Builder report output, consisting of three-up mailing labels. This is illustrated in Figure 14-16.

| | | |
|---|---|---|
| NAME 1<br>ADDRESS<br>CITY, STATE | NAME 2<br>ADDRESS<br>CITY, STATE | NAME 3<br>ADDRESS<br>CITY, STATE |

Figure 14-16 Three-Up Mailing Labels

As viewed by Dynamic Report Line Modification, each print line is 132 characters wide and extends across all three report images, regardless of the report page width of each of the images. The first print line appears to Dynamic Report Line Modification as shown in Figure 14-17.

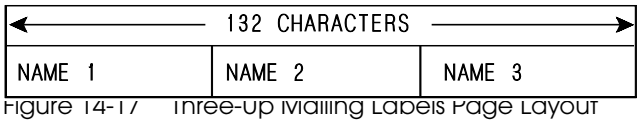



Figure 14-17 Three-up Mailing Labels Page Layout

There are two special considerations when using Dynamic Reporting with multi-image reports. The first is how to format the report preface, because that is where Dynamic Report Line Modification looks for its keyword specifications. The PREFACE must be formatted in the same way for a multi-image report as for a single image report or incorrect results will occur.

The simplest way to ensure that the PREFACE of a multi-image report is correctly formatted for Dynamic Reporting is as follows. The keywords should be coded on the first statements for the report and should fit within the first report image. Blank PREFACE lines should be added according to the following formula:

$$\text{Number of blank PREFACE lines} = [(\text{Number of print lines per report image}) \times (\text{Number of repeated images})] - (\text{Number of dynamic report line Keyword statements})$$

Figure 14-18 shows three report images as they would appear just before dynamic report line processing.

| PAGE 1 | PAGE 2 | PAGE 3 |
|---|---|---|
| .01 | .06 | .11 |
| .02 | .07 | .12 |
| .03 | .08 | .13 |
| .04 | .09 | .14 |
| .05 | .10 | .15 |

Figure 14-18 Report Page Before Dynamic Report Line Modification

Finally, Figure 14-19 shows the same three report images after dynamic report line processing using rounding.

| PAGE 1 | PAGE 2 | PAGE 3 |
|---|---|---|
| 0.0 | 0.1 | 0.1 |
| 0.0 | 0.1 | 0.1 |
| 0.0 | 0.1 | 0.1 |
| 0.0 | 0.1 | 0.1 |
| 0.1 | 0.1 | 0.2 |

Figure 14-19 Report Page After Dynamic Report Line Modification

The second special consideration is the NOSPACELINES, SPACEBEFORE, and SPACEAFTER keywords. In single image reports, Dynamic Reporting redistributes report lines on the pages as blank lines are added or eliminated. This redistribution does not take place on multi-image reports because the records appears out of sequence because dynamic reporting views the physical page not the individual image.

# Graphics

The graphics capability in VISION:Builder is a simple but effective communication tool for showing relationships between two data fields. Graphics are not currently available in VISION:Workbench for DOS or VISION:Workbench for ISPF or via ASL. They are only available in fixed format. Refere to the *VISION:Builder Reference Summary*.

The statements used to generate a graph fit into existing VISION:Builder runs as an additional report in a Request. The only restriction is that graphs cannot be produced in type P requests (preselection requests). Graphs are automatically formatted, centered, titled, dated, page numbered, and evenly scaled and labeled.

Graphs can be generated with as little as three to five statements. The output (called a graph report) consists of a graph, a detailed report, and a summary report.

There are two types of graphs.

- Scatter Diagram
- Bar Graph

Each of these graphs is defined and described in Types of Graphs on page 14-27.

Producing Scatter Diagrams on page 14-32 and Producing Bar Graphs on page 14-38 discuss the statements used to generate a scatter diagram and a bar graph, respectively.

Finally, Technical Considerations and Restrictions on page 14-44 discusses technical considerations and restrictions on the use of certain statement types while using graphs.

## Types of Graphs

VISION:Builder produces the following types of graphs:

- Scatter Diagram on page 14-28
- Bar Graphs on page 14-30

## Scatter Diagram

Figure 14-20 shows the format of a scatter diagram graph report.

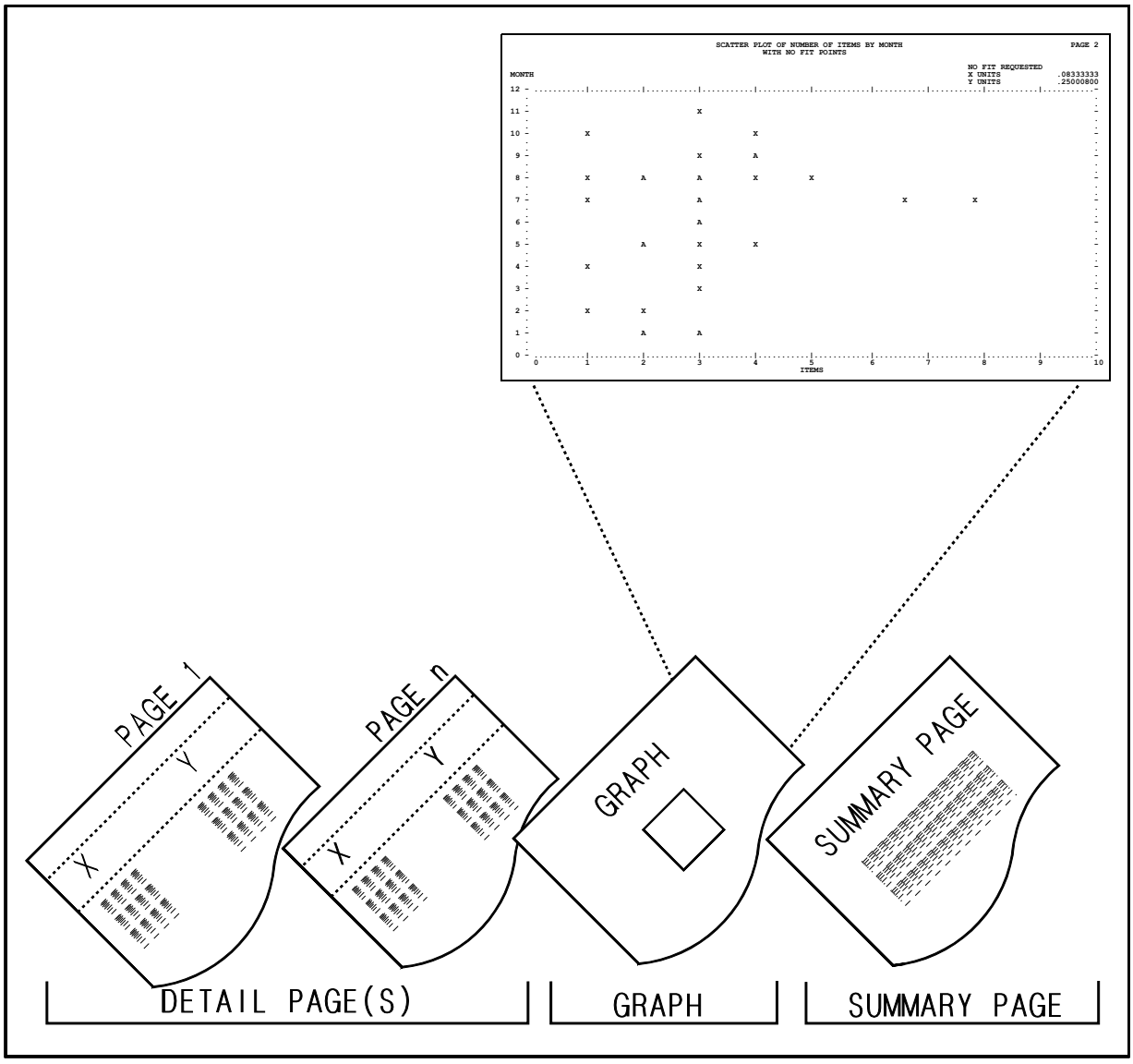



Figure 14-20 Format of a Scatter Diagram Graph Report

There are three parts to the graph report. The detail pages consist of the data values of the two fields used in the graph; it is similar to a VISION:Builder report.

The scatter diagram consists of (X,Y) ordered pairs plotted in relation to a set of perpendicular axes X and Y.

The ordered pairs consist of data values from the X field and Y field. The detail pages consist of these ordered pairs listed in columns as in a VISION:Builder report. The pairs themselves can be ordered in either ascending or descending order of X within Y. However, you have no control over the sequence of the X and Y values within each pair.

The summary page consists of statistical information such as a count of single points, a count of overlaid points, means of X and Y, standard deviation of Y (predicted Y), and so on.

Scatter diagrams can be fitted with or without a line of best prediction or least squares. The fit can be a linear least squares fit or a non-linear second order least squares fit. These lines are computed automatically by VISION:Builder if specified. They provide the best prediction of Y given X. They are widely used in statistical regression.

Details of the statements required to produce scatter diagrams can be found in Producing Scatter Diagrams on page 14-32.

## Bar Graphs

Figure 14-21 shows the format of a bar graph report.

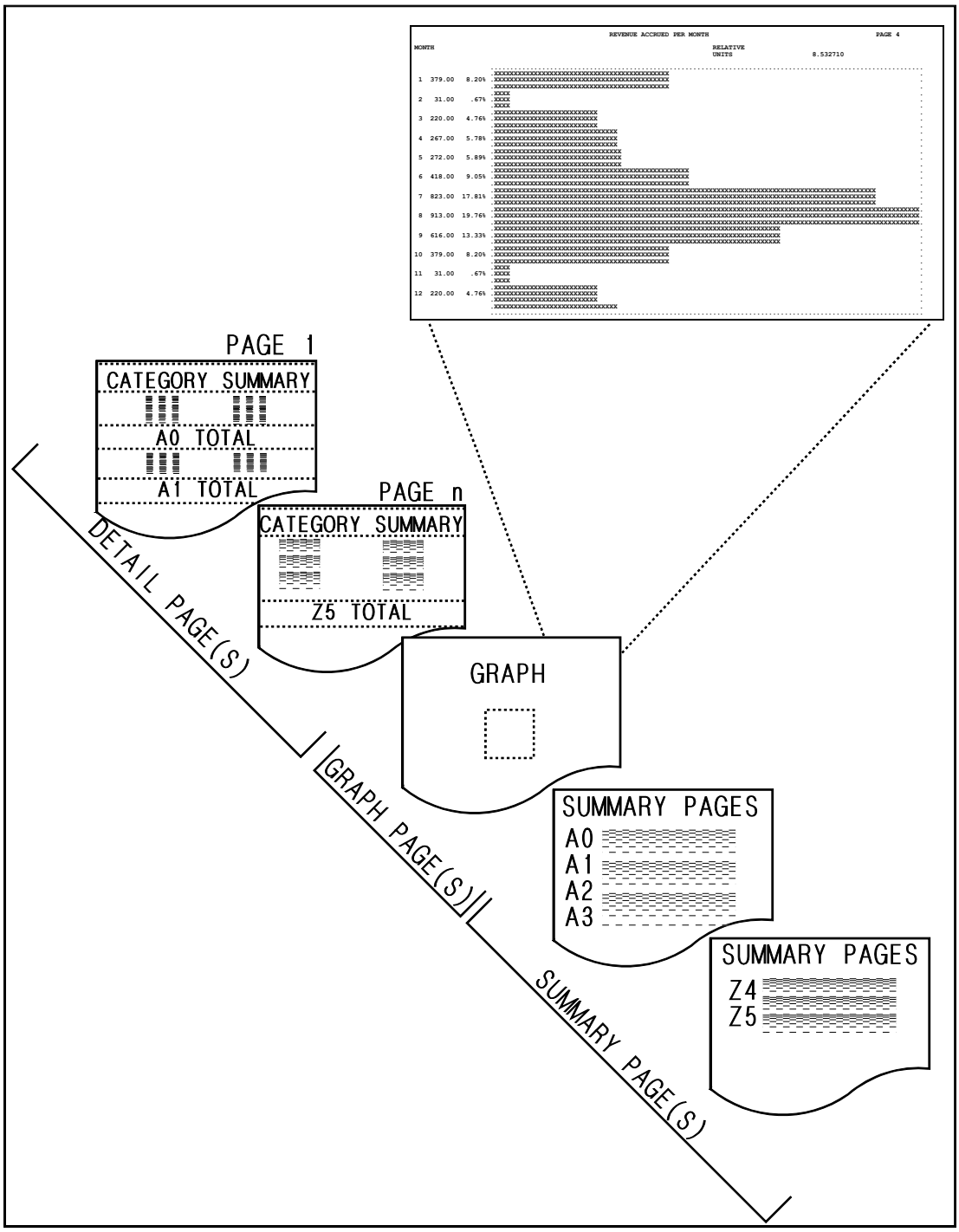



Figure 14-21 Format of a Bar Graph Report

As before, there are three parts to the graph report. The detail pages consist of the data values of the two fields used in the graph. The detail report is summarized (TOTAL, CUM, COUNT) by the field specified as the summary field.

The bar graph (or histogram) itself consists of a plot of data pairs in relation to two perpendicular axes. One value of the pair is called the summary value and the other the category value. Each point represents the summary value of a field for each category specified. The graph appears in the form of bars (either horizontal or vertical) for each category.

The two perpendicular axes in a bar graph are called the summary axis and the categorical axis, respectively. The summary axis consists of the range of summary values for a field (TOTAL, CUM, COUNT). The category axis consists of the groups or categories of a field. If the category consists of a single set of field values, it is called Discrete. If the category consists of groups of field values, it is called a Range.

Bar graphs can either be Discrete or Range graphs. The Jn statement is used to specify the type you need.

In a bar graph, a summary value for each category is plotted as a bar in relation to:

- The total of all the positive summaries or the total of all the negative summaries.
- The largest positive summary or the smallest negative summary. If you specify the former, the summary axis is said to be an absolute scale. If you specify the latter, the summary axis is said to be a relative scale.

There is no real difference between choosing the bars to be horizontal or vertical except that some summary label information is excluded in the vertical graph due to a lack of space. The important point is to correctly specify the fields to be categorized and summarized and to specify whether the graph is to be Discrete or Range.

The summary pages consist of useful information, such as category start and end values, counts of invalid and missing data, counts of summary values for each category, and percentages of summary values for each category.

Details of the statements required to produce bar graphs can be found in Producing Bar Graphs on page 14-38.

## Producing Scatter Diagrams

A minimum of five statements is required for producing a scatter diagram. They are as follows:

- Jn statement graph type (positions 11 and 12) = SN, SL, or SS
- Kn statement graph modifier (position 11) = X
- Kn statement graph modifier (position 11) = Y
- Rn statement graph modifier (position 37) = X
- Rn statement graph modifier (position 37) = Y

The Jn statement for a graph is analogous to the En statement for a request. It is known as the graph specification statement. It functions as a formatting statement for the entire graph by specifying the type of graph, the kind of fit plot character, and so on.

The Kn statement or axis specification is analogous to the Rn statement for a request. It defines the axes or elements of the graph by specifying the starting and ending points of each axis, the increment (delta) for each axis, and so on.

The Rn statements function in the usual manner by specifying the two fields being used to generate the graph. There are, however, restrictions on certain entries in the Rn statement; these are described in Technical Considerations and Restrictions on page 14-44.

These five statements can be used along with other VISION:Builder statements to make up a request (except a type P request). Again, there are restrictions on some of the entries in some of the statements; these are described in Technical Considerations and Restrictions on page 14-44.

Figure 14-22 shows the scatter diagram from a typical graph.

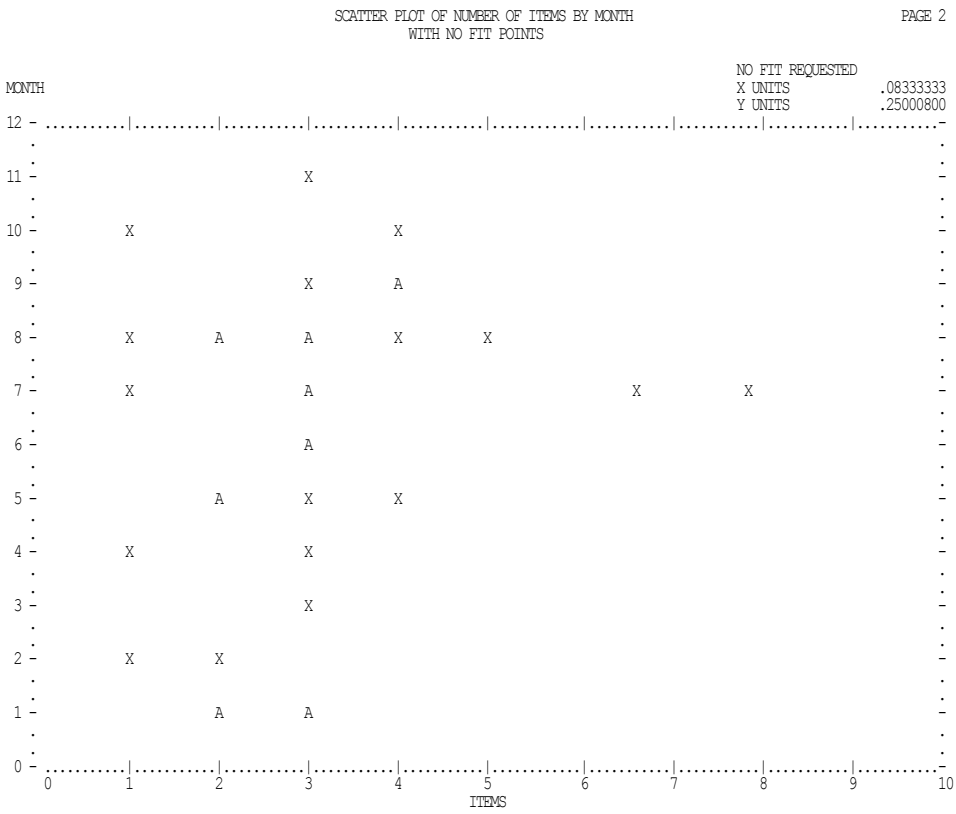



Figure 14-22 Scatter Diagram with No Fit

The X-axis represents the number of items ordered. The Y-axis represents the field MONTH. Each point on the graph represents the ordered pair of values for the number of items and the month. The letter X represents a single ordered pair and the letter A represents multiple ordered pairs. X units = nnnn tells you how much each print position on the X-axis represents, similarly for Y units.

This scatter diagram requests no fit plot points to be printed for the prediction lines.

The A represents two or more overlapping points. It is specified as the secondary plot character in position [18] of a Jn statement. Figure 14-23 shows how the scatter diagram was generated using only seven statements.

```
                                                  ******************************
                                                  *  REQUEST NAME - TEST       *
                                                  *  INPUT STREAM REQUEST      *
                                                  ******************************
-------------------------------------------------------------------------------------------------------------------
              SUPP                 PRIM  SECN  FIT   HORZ  HORZ  VERT  VERT
STMT  GRAPH   DATA    BAR    BAR   PLOT  PLOT  PLOT  AXIS  HASH  AXIS  HASH
TYPE  TYPE   REPORT   TYPE  WIDTH  CHAR  CHAR  CHAR  CHAR  CHAR  CHAR  CHAR
-------------------------------------------------------------------------------------------------------------------
(J1)  (SN)                                (A)
-------------------------------------------------------------------------------------------------------------------
STMT  GRAPH  ................. INTERVAL CONTROL .......................
TYPE  MODFR  ......START.......  .......END........  ......DELTA.......
-------------------------------------------------------------------------------------------------------------------
(K1)   (X)   (0              )  (10             )  (1              )
(K1)   (Y)   (0              )  (12             )  (1              )
-------------------------------------------------------------------------------------------------------------------
         SP  Q                E  N  S  D  C  S  T  C  C  M  M  A  G  D Q PCT/RAT     P                PARTIAL
STMT SEQ  BF  L  FIELD   SEG-  N  P  E  E  T  U  O  U  N  A  I  V  M  E L  FIELD   RA- C     PICTURE    STRT NO
TYPE NO.  COL F  NAME     LVL  D  R  Q  S  L  B  T  M  T  X  N  G  R  C F  NAME    TIO T      EDIT      CHR CHR
-------------------------------------------------------------------------------------------------------------------
(R1)         (T,ITEMS   )                                        (X)
(R1)         (T,MONTH   )                                        (Y)
-------------------------------------------------------------------------------------------------------------------
STMT  SEQ
TYPE  NO.                                                 TEXT
-------------------------------------------------------------------------------------------------------------------
(T1) (010)                  (SCATTER PLOT OF NUMBER OF ITEMS BY MONTH#                )
(T1) (010)                  (WITH NO FIT POINTS#                                      )
```

Figure 14-23 Source Statements for a Scatter Diagram

The J1 statement uses an SN for scatter diagrams with no fit plot points. The secondary plot character has been changed to an A. The default is an asterisk (*). The axis characters have been changed to a period (.), instead of a dash (-). The first K1 statement specifies a start of zero and end of ten, increasing by one. The K1 statements specify the intervals on the X and Y axes. The report statements list the fields and relate them to the axes by means of the X and Y in the graph modifier, position 37 of an Rn statement.

Scatter graphs can be generated from type Z (zoned), P (packed), or F (fixed) fields only. For more information about field definitions, see Chapter 4, *Field Characteristics and Handling*.

Figure 14-24 shows the same scatter diagram with linear fit plot points. A linear fit plot is a line through the scatter diagram computed automatically by VISION:Builder that best predicts a Y value given an X value.

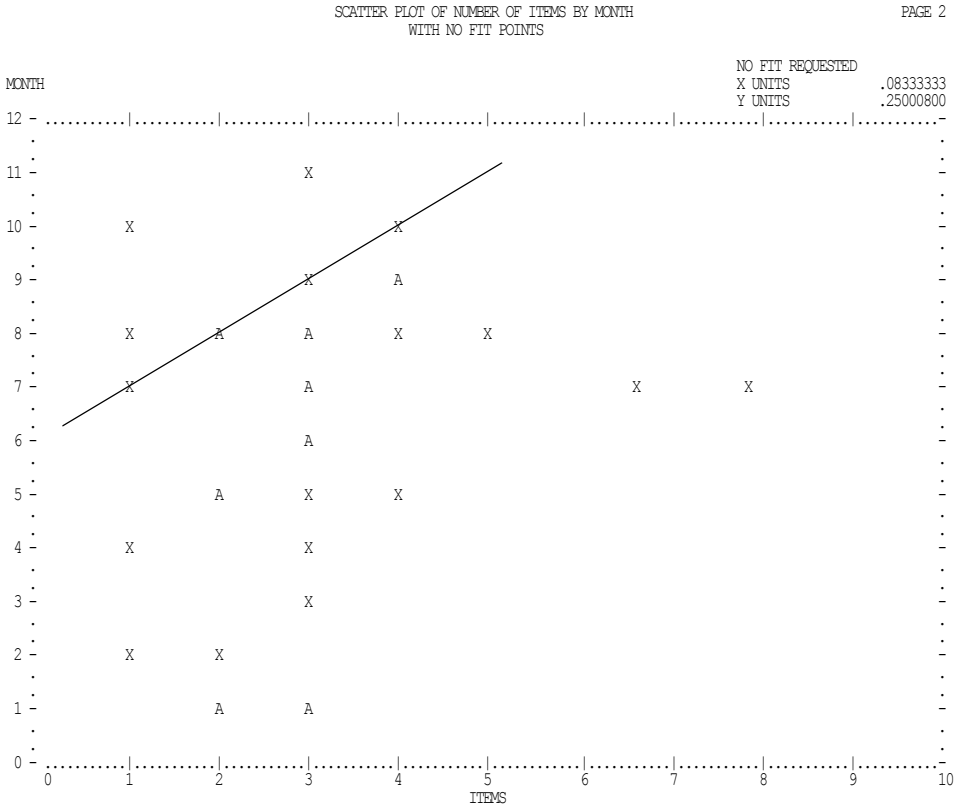



Figure 14-24 Scatter Diagram with Linear Fit Plot

The linear fit plot points are printed using period (.) characters as the default. This graph is generated by specifying SL instead of SN in the Jn statement positions 11 and 12. Specifying an SS produces secondary fit plot points.

By default, a detailed report and a summary page are always produced with any graph. The detailed report (graph data report) can be suppressed using a Y in position 13 of a Jn statement.

Figure 14-25 shows the detailed report for this scatter diagram. It lists the ordered pairs in ascending sequence by X and then Y within X.

```
                                    SCATTER PLOT OF NUMBER OF ITEMS BY MONTH                          PAGE 1
                                              WITH NO FIT POINTS
-------------------------------------------------------------------------------------------------------------------------------
 ITEMS  MONTH
-------------------------------------------------------------------------------------------------------------------------------
    1      2
    1      4
    1      7
    1      8
    1     10
    2      1
    2      1
    2      2
    2      5
    2      5
    2      8
    2      8
    2      8
    2      8
    3      1
    3      1
    3      3
    3      4
    3      5
    3      6
    3      6
    3      6
    3      7
    3      7
    3      7
    3      7
    3      7
    3      7
    3      8
    3      8
    3      9
    3     11
    3     12
    4      5
    4      8
    4      9
    4      9
    4     10
    5      8
    7      7
   11      8
```

Figure 14-25 Detail Report for Scatter Diagram

Figure 14-26 shows the summary page for this scatter diagram. An invalid point occurs if any one of the ordered pairs becomes invalid (see Chapter 4, *Field Characteristics and Handling*). A missing point occurs if any one of the ordered pairs is missing or unavailable. In both cases, the point is rejected from the graphics output.

```
                                    SCATTER PLOT OF NUMBER OF ITEMS BY MONTH                          PAGE 3
                                              WITH NO FIT POINTS

COUNT OF SINGLE POINTS                    16
COUNT OF OVERLAID POINTS                   8
COUNT OF OUT OF RANGE POINTS               2
COUNT OF INVALID/MISSING POINTS            0
COUNT OF ALL POINTS                       41

MEAN X                          2.794859886
MEAN Y                          6.230754852

STANDARD DEVIATION              2.674295425

LINEAR FIT EQUATION          Y =  .51272273 X + 4.797766685
STANDARD ERROR OF ESTIMATE      2.67701435

SECOND ORDER FIT EQUATION    Y =  .067907691 X**2 + 5.609140396
STANDARD ERROR OF ESTIMATE      2.684913635
```

Figure 14-26 Summary Page for Scatter Diagram

The summary statistics can be interpreted as follows:

The mean of X is given by $\frac{\Sigma\ Xi}{n}$

Where: i = 1 to n

n = number of valid points

$\Sigma$ = sum

The mean of Y is given by $\frac{\Sigma\ Yi}{n}$

The standard deviation of Y is given by $\sqrt{\frac{\Sigma\ (Yi - \bar{Y})^2}{n}}$

Where: $\overline{Y}$ = mean of Y

$\sqrt{}$ = square root

**Note:** If multiple graphs are required each time the value of a third field changes, the third field is called a graph control field. Any field can be used as a graph control field by specifying a G in position [37] of an Rn statement. The detailed report is grouped by this field.

The linear prediction equation is in the form: Y = a + bX

Where: a = intercept constant

b = slope, given by $\frac{\iota\ \Sigma\ XY\ -\ \Sigma\ X\ \Sigma\ \lambda}{n\ \Sigma\ X^2 - (\Sigma\ X)^2}$

The second order least squares fit equation is in the form: Y = a + bX

The standard error of estimate is given by $\mathfrak{S}E\ =\ \sqrt{\frac{\Sigma\ (Yi - Yi')^2}{n-2}}$

Where: Yi' = predicted value of Yi given Xi

## Producing Bar Graphs

A minimum of three statements is required for producing a discrete bar graph. They are as follows:

- Jn statement graph type (positions 11 and 12) = HD or VD
- Rn statement graph modifier (position 37) = S
- Rn statement graph modifier (position 37) = C

A minimum of four statements is required for producing a range bar graph. They are as follows:

- Jn statement graph type (positions 11 and 12) = HR or VR
- Kn statement graph modifier (position 11) = C
- Rn statement graph modifier (position 37) = S
- Rn statement graph modifier (position 37) = C

The Jn or graph specification statement specifies the type of graph (HD for horizontal discrete, HR for horizontal range, and so on). It also describes the width of the bar, the kind of fit plot character, and so on.

The Kn statement or axis specification defines the category axis of a range bar graph. It is not needed for discrete bar graphs. The delta or increment specifies the extent of the range bar graph. For example, a delta of three for the category axis consisting of 12 months will give you four bars, one for each quarter.

The Rn statements are used to specify which fields represent the category axis and which one represents the summary. It also specifies the kind of summary (TOTAL, CUM, or COUNT).

These statements can be used along with other VISION:Builder statements to make up a request (except a type P request). Again, there are restrictions on some of the entries in some of the statements; these are described in Technical Considerations and Restrictions on page 14-44.

Figure 14-27 shows a discrete bar graph from a typical graph report.

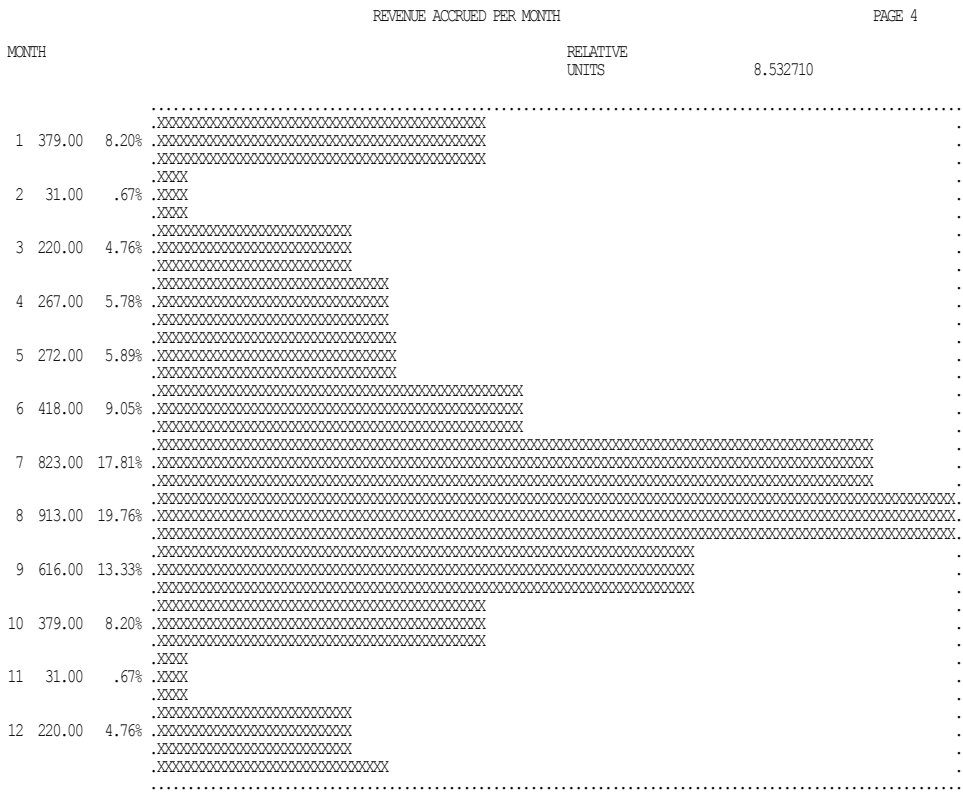



Figure 14-27 Discrete Bar Graph

In this graph, the summary axis is horizontal (so the graph is called horizontal). The field ITMPRICE is summarized by taking a total. The category axis is vertical and represents the field MONTH. Each bar is three units wide and represents the relative total price of items for each month.

This bar graph has a relative scale. This means that the length of each bar is computed relative to the length of the longest bar.

Each X print character in the bar represents 8.532710 in this case. The percentages printed along the category axis represent the summarized value for that category relative to the total summarized values for all categories. Negative percentages represent the negative summary value relative to the total negative summary values for all categories.

The second column printed along the category axis shows the summary value for that category. In this case, it shows the total price of items for each month. It represents the length of the bar.

Detailed pages and a summary page are also produced for each graph by default. The detailed report can be overridden using a Y in position [13] of a Jn statement.

Figure 14-28 shows the detailed report for this discrete bar graph. It lists the detail values of the field being summarized, grouped by each category (in this case, each month). The summary value for each category is also printed. Each category is automatically assigned a label (A0 through Z5). The maximum number of categories is 256. If the label prints with a dash (-), it means the contents of that category are missing. If the label prints with an asterisk (*), it means the contents of that category are invalid.

```
                                                  REVENUE ACCRUED PER MONTH                                    PAGE 1
-------------------------------------------------------------------------------------------------------------------------------
                    MONTH
                              PRICE
-------------------------------------------------------------------------------------------------------------------------------
                      1       $5.00
                                5.00
                                5.00
                               15.00
                               15.00
                               23.00
                               23.00
                               28.00
                              100.00
                              160.00
MONTH    TOTAL A0     1     $379.00
                      2        $.00
                                3.00
                               28.00
MONTH    TOTAL A1     2      $31.00
                      3      $20.00
                              100.00
                              100.00
MONTH    TOTAL A2     3     $220.00
                      4      $15.00
                               25.00
                              100.00
                              127.00
MONTH    TOTAL A3     4     $267.00
                      5       $3.00
                                5.00
                                6.00
                               20.00
                               20.00
                               22.00
                               22.00
                               24.00
                               30.00
                               60.00
                               60.00
MONTH    TOTAL A4     5     $272.00
                      6       $4.00
                                5.00
                               20.00
                               22.00
                               24.00
                               28.00
                               30.00
```

Figure 14-28 Detail Report for Discrete Bar Graph

Figure 14-29 shows the summary page produced with a discrete bar graph. Most of the entries are self-explanatory. The percentages are computed by taking a percentage of each summary value relative to the total of all summary values.

REVENUE ACCRUED PER MONTH PAGE 5

| ID | CATEGORY VALUE | COUNT OF DATA VALUES VALID | INVALID | MISSING | TOTAL | SUMMARY VALUES POSITIVE | NEGATIVE | PERCENTAGE VALUES POSITIVE | NEGATIVE |
|---|---|---|---|---|---|---|---|---|---|
| A0 | 1 | 10 | 0 | 0 | 10 | 379.00 | | 8.20 | |
| A1 | 2 | 3 | 0 | 0 | 3 | 31.00 | | .67 | |
| A2 | 3 | 3 | 0 | 0 | 3 | 220.00 | | 4.76 | |
| A3 | 4 | 4 | 0 | 0 | 4 | 267.00 | | 5.78 | |
| A4 | 5 | 11 | 0 | 0 | 11 | 272.00 | | 5.89 | |
| A5 | 6 | 9 | 0 | 0 | 9 | 418.00 | | 9.05 | |
| A6 | 7 | 26 | 0 | 0 | 26 | 823.00 | | 17.81 | |
| A7 | 8 | 35 | 0 | 0 | 35 | 913.00 | | 19.76 | |
| A8 | 9 | 11 | 0 | 0 | 11 | 616.00 | | 13.33 | |
| A9 | 10 | 5 | 0 | 0 | 5 | 260.00 | | 5.63 | |
| B0 | 11 | 3 | 0 | 0 | 3 | 87.00 | | 1.88 | |
| B1 | 12 | 3 | 0 | 0 | 3 | 335.00 | | 7.25 | |
| TOTALS | | 123 | 0 | 0 | 123 | 4621.00 | | | |

Figure 14-29 Summary Page for Discrete Bar Graph

If multiple graphs are required for each change in value of a third field (graph control field), use an Rn statement using the field name and a G in position [37]. The detailed report is subtitled by each value of G.

Figure 14-30 shows how the discrete bar graph was generated.

```
                                    ****************************
                                    *  REQUEST NAME - TEST     *
                                    *  INPUT STREAM REQUEST    *
                                    ****************************

STMT  GRAPH  SUPP DATA  BAR   BAR   PRIM PLOT  SECN PLOT  FIT PLOT  HORZ AXIS  HORZ HASH  VERT AXIS  VERT HASH
TYPE  TYPE   REPORT     TYPE  WIDTH CHAR       CHAR       CHAR      CHAR       CHAR       CHAR       CHAR
(J3)  (HD)             (R)   ( 3)                       (.)                   (.)

STMT SEQ  SP BF COL  Q L F  FIELD NAME  SEG- LVL  E N P D R  S E Q  D E T S  C T U L  S T O B  T C U M  C N T  M A I X  A V G N  G M R  D E C  Q L F  PCT/RAT FIELD  RA- TIO  P C T  PICTURE EDIT  PARTIAL STRT CHR  NO CHR
(R3)          (T,MONTH     )                                                                                                                                  (C)
(R3)            (ITEMPRICE)                                                                                                                                  (S)

STMT  SEQ
TYPE  NO.                                                                TEXT
(T3)  (010)                (REVENUE ACCRUED PER MONTH ITEMS BY MONTH#                )
```

Figure 14-30 Source Statement for Discrete Bar Graph

The J3 statement uses an HD keyword to specify a horizontal discrete bar graph. The bar type of R specifies a relative scale for the summary axis. The periods (.) in positions [20] and [22] specify periods instead of dashes as the print character for the axes.

The first R3 statement specifies MONTH to be the category axis and the second R3 specifies ITMPRICE with a total summary to be the summary axis.

Discrete bar graphs can use type Z (zoned), P (packed), F (fixed), or C (character) fields for their axes. For more information about field definitions, see Chapter 4, *Field Characteristics and Handling*.

Figure 14-31 shows the same bar graph as shown in Figure 14-27 but as a range bar graph.

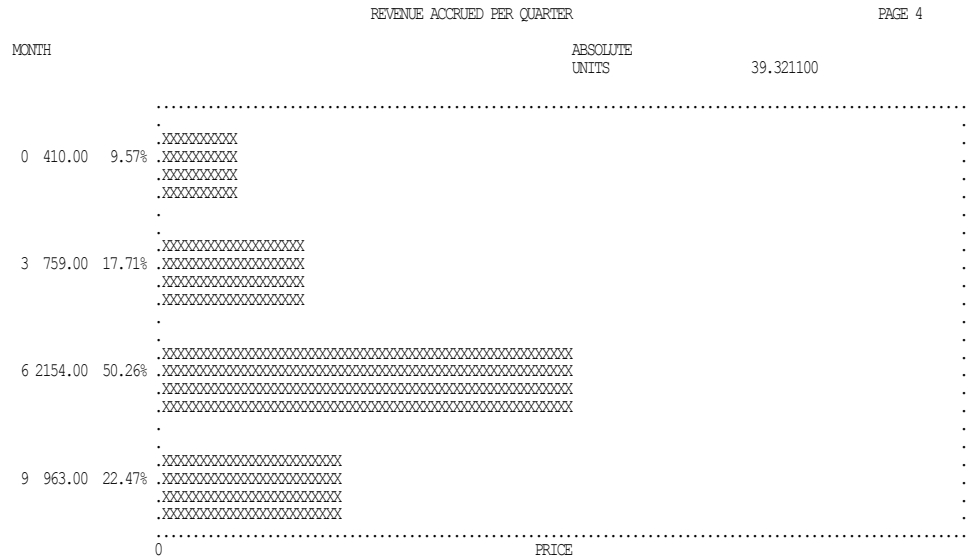



Figure 14-31 Range Bar Graph

The only difference between this graph and the discrete graph is that the category axis is now grouped into ranges instead of being left discrete. In this graph, the category axis is divided into four categories or bars. Each bar represents three months instead of one month as before.

Note that this graph also uses absolute units for its summary axis scale. Each X in the bar represents 39.321100 dollars. The length of the bar is given by the second column of numbers along the category axis. The percentage represents the summary value for the category relative to the total of the summary values for all the categories.

The detailed pages and summary page for this bar graph are similar to the discrete bar graph and are not shown here.

Figure 14-32 shows how the range bar graph was generated using a total of five statements.

```
                                                  ****************************
                                                  *  REQUEST NAME - TEST     *
                                                  *  INPUT STREAM REQUEST    *
                                                  ****************************
-------------------------------------------------------------------------------------------------------------
              SUPP                 PRIM  SECN  FIT   HORZ  HORZ  VERT  VERT
STMT  GRAPH   DATA    BAR    BAR   PLOT  PLOT  PLOT  AXIS  HASH  AXIS  HASH
TYPE  TYPE   REPORT   TYPE  WIDTH  CHAR  CHAR  CHAR  CHAR  CHAR  CHAR  CHAR
-------------------------------------------------------------------------------------------------------------
(J4)  (HR)            (A)    ( 4)                     (.)         (.)
-------------------------------------------------------------------------------------------------------------
STMT  GRAPH  ................. INTERVAL CONTROL .......................
TYPE  MODFR  ......START.......  .......END........  ......DELTA.......
-------------------------------------------------------------------------------------------------------------
(K4)   (C)   (0              )  (12             )  (3              )
-------------------------------------------------------------------------------------------------------------
         SP  Q                E  N  S  D  C  S  T  C  C  M  M  A  G  D Q PCT/RAT     P                PARTIAL
STMT SEQ  BF  L  FIELD   SEG-  N  P  E  E  T  U  O  U  N  A  I  V  M  E L  FIELD   RA- C     PICTURE    STRT NO
TYPE NO.  COL F  NAME     LVL  D  R  Q  S  L  B  T  M  T  X  N  G  R  C F  NAME    TIO T      EDIT      CHR CHR
-------------------------------------------------------------------------------------------------------------
(R4)         (T,MONTH    )                                          (C)
(R4)            (ITMPRICE)                        (X)              (S)
-------------------------------------------------------------------------------------------------------------
STMT  SEQ
TYPE  NO.                                                 TEXT
-------------------------------------------------------------------------------------------------------------
(T4)  (010)                  (REVENUE ACCRUED PER QUARTER#                                )
```

Figure 14-32 Source Statements for Range Bar Graph

The J4 statement specifies a horizontal range graph, positions [11] and [12]. The scale chosen for the summary axis is absolute, position [14]. The width of the bar is four and the axis print characters are periods (.).

The K4 statement must be used for a range bar graph. It is used to specify the ranges for the category axis (C in position [11] for category axis). The delta increment of three divides the category axis into four bars.

The R4 statements specify the fields to be used as the category axis and summary axis. The summary chosen is total for the ITMPRICE field.

Range bar graphs can only use field types Z, P, and F for the category axis and field types Z, P, F, and C for the summary axis.

Multiple graphs can be produced for each value of a graph control field in the same way as for a discrete bar graph.

## Technical Considerations and Restrictions

### Considerations

- A report is a graph report if it contains a Jn and/or Kn statement.
- If there are multiple Jn statements for a report, the last one in the input stream is used. The others are ignored.
- A type C field is converted to numeric for range bar graphs but is treated as a character string for discrete bar graphs during the report step. During the sort it is treated as a character string.
- Floating point option must be available to calculate (and plot) fit equations and mean.
- Graph reports are available with and without RPO.
- Maximum number of categories is 256.
- Each graph field should be referenced as the result field of a PR operation to ensure the correct sort sequence for invalid fields; otherwise, the invalid fields will not always print at the bottom of the detailed report.
- Minimum page width = 120.
- Minimum page height = 6 for scatter diagram without fixed points, 16 for scatter diagram with fixed points, number of title lines plus subtitle line plus 4 detail lines for bar graphs.
- Graphs cannot be used in type P requests.
- Graphs cannot be used with formatted reporting or formatted sectional reporting.

## Restrictions on the Use of Other Statements

| Statement | Restrictions | | |
|---|---|---|---|
| En | summary report only | | Not allowed. |
| | vertical spacing | | |
| | number of repeated images | | |
| | report handling = NR or DN | | |
| | column heading position | | Data report only. |
| | line numbers | | |
| | label on summary lines | | |
| ER | summary report only | | Not allowed. |
| | vertical spacing | | |
| | line numbers | | Data report only. |
| RC | sort control = X or Y | | Not allowed. |
| Rn | sequence | max | Ignored. |
| | descending | min | |
| | control | avg | |
| | subtitle | ratio | |
| Fn | | | Not allowed. |

## Increasing Precision

To increase the precision of your graph's scales, the axes should be defined to have smaller units. Each point on the axes would represent a smaller number of units.

For a bar graph, the precision can be increased by increasing the page width or page height because units are computed using one of the two formulas:

$$\text{Absolute Units} = \frac{\text{sum of all summaries}}{\text{page width or page height}}$$

$$\text{Relative Units} = \frac{\text{largest summary value}}{\text{page width or page height}}$$

For a scatter diagram, the precision can be increased by increasing the page width or page height or by decreasing the absolute difference between the start and end values:

$$\text{Units} = \frac{\text{end} - \text{start}}{\text{page width or height}}$$

## Minimum Plotted Points for Scatter Diagram Calculations

To calculate fit plots, mean of X and Y, standard deviation, and standard error of estimate, the following minimum number of valid points are needed.

| Calculation | Minimum Requirement |
|---|---|
| Mean X and Y | 1 valid plotted point |
| Standard deviation | 2 valid plotted points |
| Least squares linear fit | 2 valid, unique points plotted |
| Standard error of estimate for linear fit | 2 valid, unique points plotted |
| Second order least squares fit | 3 valid, unique points plotted |
| Standard error of estimate for second order fit | 3 valid, unique points plotted |

If any of these minimums are not provided in the data, NOT AVAILABLE is printed on the summary page.

## Floating Point Arithmetic and Graphics Operation

The operating characteristics of floating point arithmetic affect graphics output in two ways. One concerns the slope of fit plots on scatter diagrams and the equations depending on the slope value. The other concerns the loss of significance in the mean of X and Y, standard deviation, linear and second order fit equations, and standard error of estimate. The loss of accuracy that occurs when floating point conversions are made is the reason for these effects.

When the linear or second order fit plot of a scatter diagram has a slope that is vertical or near vertical (all X values are the same or nearly the same), the resulting fit plot equations can be reported as NOT AVAILABLE on the summary page. In some cases, equations are given but they are unreliable. The same effects can occur in the standard deviation and standard error of estimate for the scatter diagram; equations can be either NOT AVAILABLE or not reliable.

If a scatter diagram produces a summary page with these features, a vertical or near vertical slope may be the explanation.

Under certain conditions, graphics print the message SIGNIFICANCE LOSS next to the label for the mean of X and Y, standard deviation of Y, standard error of estimate, and linear and second order fit equations. This message is printed when packed numbers with more than 15 integer digits are encountered. These packed

numbers are converted to floating point values for calculation; loss of significance occurs because some accuracy is lost when converting the floating point number to character format for printing.

## How Graphics Page Layouts Are Calculated

Graphics page layouts are calculated using the formula below.

Minimum page height specified on the En or ER statement must be equal to or greater than:

A + B + C + the greatest of (X or Y or Z)

The elements of this formula are illustrated in Figure 14-33 and explained in the following text.

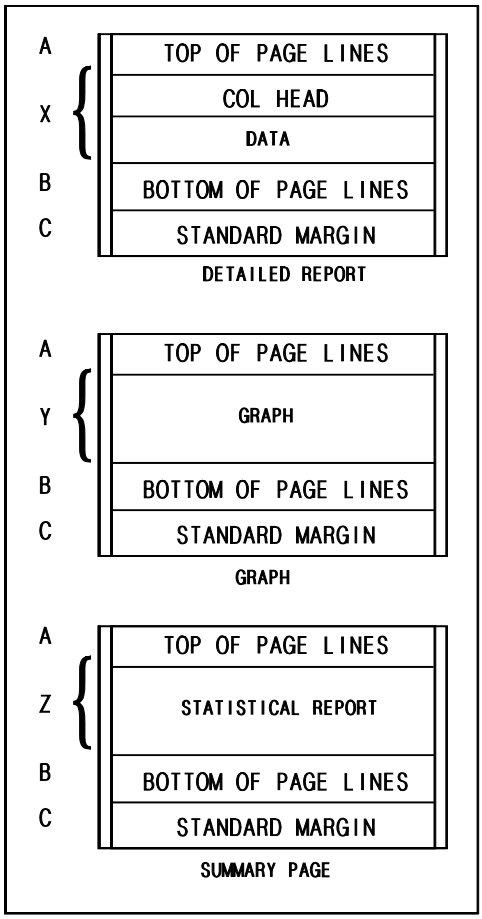



Figure 14-33 Graphics Page Layout

A = top of page lines

title lines
+ graph control field (subtitle), if any
+ page number and date
+ blank lines required

B = bottom of page lines

title lines
+ page number and date

C = 6 lines (standard margin)

X = column headings + borders + 1 detail line

Y = 3 lines or the number of column heading lines, whichever is greater

+ the length of the vertical axis
+ label lines
+ the number of column heading lines for the horizontal axis
+ 1 blank line

Z = 6 lines in a bar graph or in a scatter diagram without floating point option
or 16 lines in a scatter diagram with floating point option

### The Length of the Vertical Axis

The length of the vertical axis is calculated according to the following criteria:

For a vertical bar graph:

Length = { Two for positive or negative summaries / or / Four for positive and negative summaries } + Sufficient space to permit values of data to be shown by bar heights. One printline is the legal minimum, but it will not yield a very useful graph

For a horizontal range bar graph:

LENGTH = the number of intervals + 2

For a horizontal discrete bar graph:

LENGTH = the number of categories to be graphed + 2

For a scatter diagram:

LENGTH = the number of intervals + 1

### The Number of Label Lines

The number of label lines for each graph type is as follows:

| | | |
|---|---|---|
| Scatter diagrams | 2 lines | 1 for the label, 1 blank |
| Horizontal bar graphs | 2 lines | 1 for the label, 1 blank |
| Vertical bar graphs | 3 lines | 2 for the label, 1 blank |

## Painting a Report Section in VISION:Workbench for DOS

Report painting is designed for situations where the standard report format is acceptable but you want to customize just the report title, column headings, or summary areas. VISION:Workbench for DOS provides you with the capability of painting the report sections you want while accepting the standard report layout for the rest of the report. It is analogous to Formatted Sectional reports available in both ASL and fixed format VISION:Builder syntax.

With VISION:Workbench for DOS, you can selectively paint one or more of the following sections of a report: the title, column heading, and summary sections. Any section you do not choose to paint is formatted according to the defaults for a standard VISION:Builder report.

Painting a section of a report is especially useful when reports require data values in the title or column headings or when custom labeling and placement of summaries is wanted. In addition, you can also perform arithmetic calculations on the summaries that VISION:Builder has computed and report the results in the summary section.

The topic of painting a report section is described in the *VISION:Workbench for DOS Reference Guide*. A Report Summaries window is shown in Figure 14-34 as a sample of painting the summaries in a report. Notice that the variable information can be placed anywhere in the summary area by placing underscores where you want the variable information to appear. The Contents popup window displays the fields whose contents will print in place of the underscores. A summary value computed by VISION:Builder or an expression involving fields and summary values can also be entered in the Contents popup window.

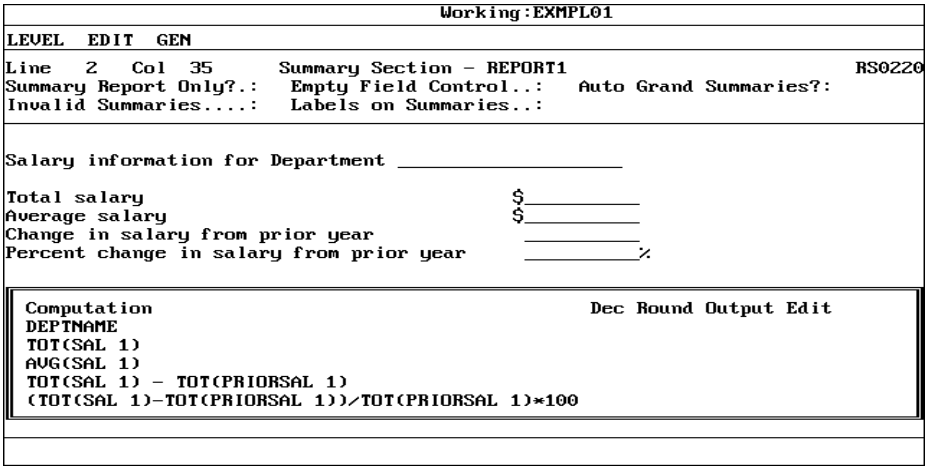



Figure 14-34 Painting Summary Reports

Chapter

# 15 Advanced Transaction Processing

**Note:** This chapter only applies to the VISION Builder 4000 model series.

In Chapter 10, *Transactions*, transaction definitions cataloged in the common library maintained the master file using just one statement in the updating application run. This is satisfactory if your application transaction records are closely related to the master file and no additional input processing or calculations are required. Complex applications, however, often require extensive editing of the transaction records and reference indexed or user coordinated files to support the maintenance of the master file.

Figure 15-1 (repeated from Chapter 2, *Application Development System*) presents a high level overview of the transaction processing cycle. This chapter discusses how to insert procedures into the transaction cycle, what types of events within the cycle trigger these procedures, and the processing actions available to each type of procedure.

Be sure to read Chapter 10, *Transactions* before you read Chapter 15.

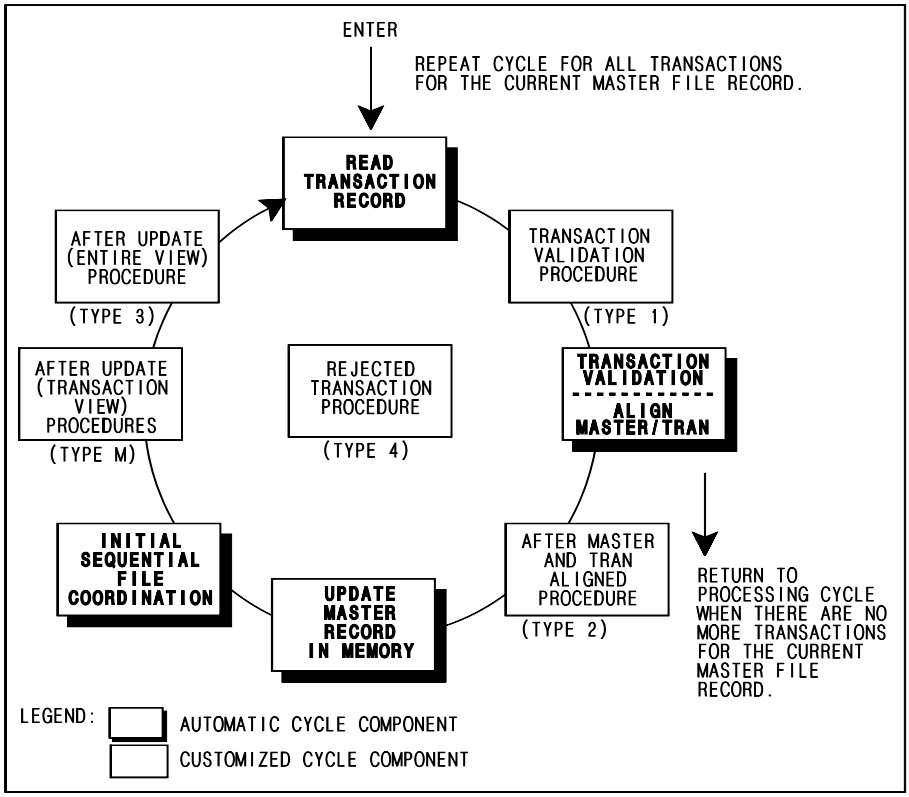



Figure 15-1 Transaction Cycle

Within the Transaction Cycle, there are points where transaction procedures can be inserted to extend the updating process. These transaction procedures can perform a variety of tasks to tailor and control the transaction cycle and update process in order to accommodate any specific processing requirements.

Transaction procedures are specified with a procedure type designation that triggers the timing of their execution within the transaction cycle. The various transaction procedure types are discussed in Transaction Procedures on page 15-6. As shown in Figure 15-1, the transaction procedure Types are 1, 2, M, 3, and 4. There are two additional transaction procedure Types known as 0 (zero/INIT) and 5 (pre master read). These procedures are the customized cycle components of the transaction cycle.

It is important to understand that each transaction procedure type executes at a specific point in time within the transaction cycle. These points are positioned around the automatic cycle components and events that occur during the transaction processing cycle.

# Transaction Processing Step

Figure 15-2 shows a transaction application processing step with all the elements that can be utilized by a transaction processing application. The use of any and all of the elements within an application is easily specified in the application run control group and procedures. To reference the various work areas, you must code an associated file definition for each area. This is accomplished in a separate definition run prior to the application processing run.

In this chapter we talk about work areas being available. You can reference any work area at any time by using the appropriate qualifier. When there is data in the work area and it is available to a procedure, the data can be referenced and manipulated. Where data does not exist or when a work area is unavailable, references will result in a missing (non-existent) condition. Manipulation of missing data fields will produce invalid results. See Invalid, Missing, and Overflow Fields on page 4-18 for more information.

Figure 15-3 shows the transaction cycle flow in a flowchart format. The flowchart combines the automatic components and transaction procedures of the transaction processing cycle. Included are the decision diamonds that direct the flow during the cycle. Also included are the points where the Normal (Type N) and EOF (Type E) procedures would be executed. The Normal procedures are executed after the transaction cycle has completed for a specific master record. The EOF procedure is executed when the transaction and master files are at end-of-file.

The First Round Coordination boxes apply to sequential coordination files. This is discussed in Chapter 6, *Input File Processing*. Within the transaction cycle, the initial alignment between the master file and the associated sequential coordinated files is performed here.

Note that in Figure 15-3, TF is the transaction file record key value, and MF is the master file record key value

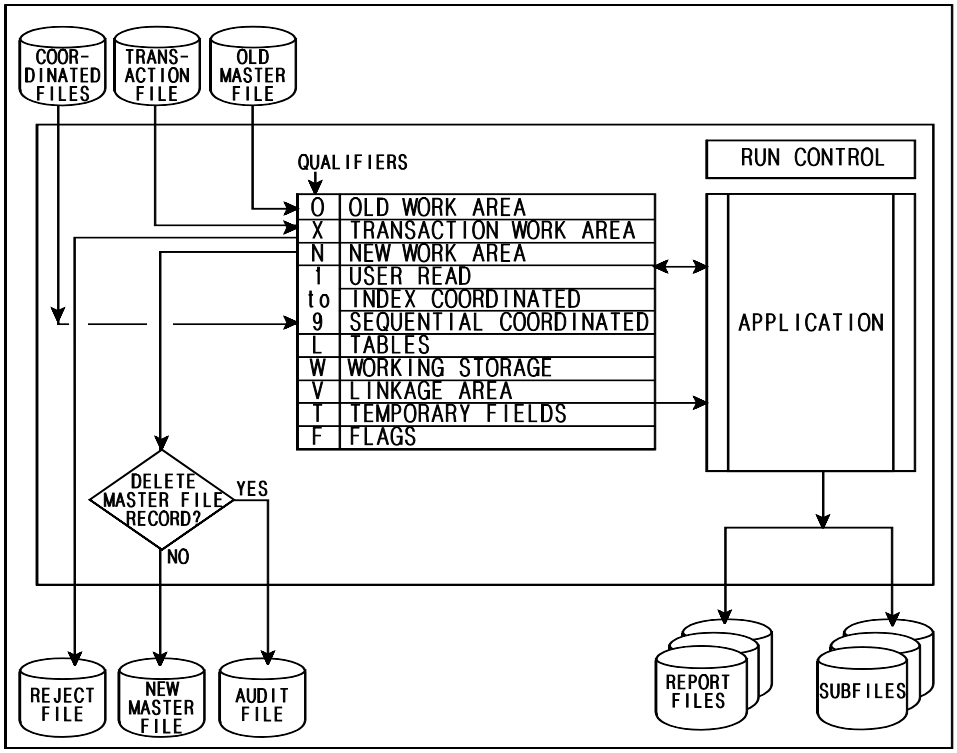



Figure 15-2 Transaction Processing Step Suitable for Transaction Requests and Coordinated Files

**Note:** In Figure 15-2, old and new master files are the same physical file if updating-in-place.

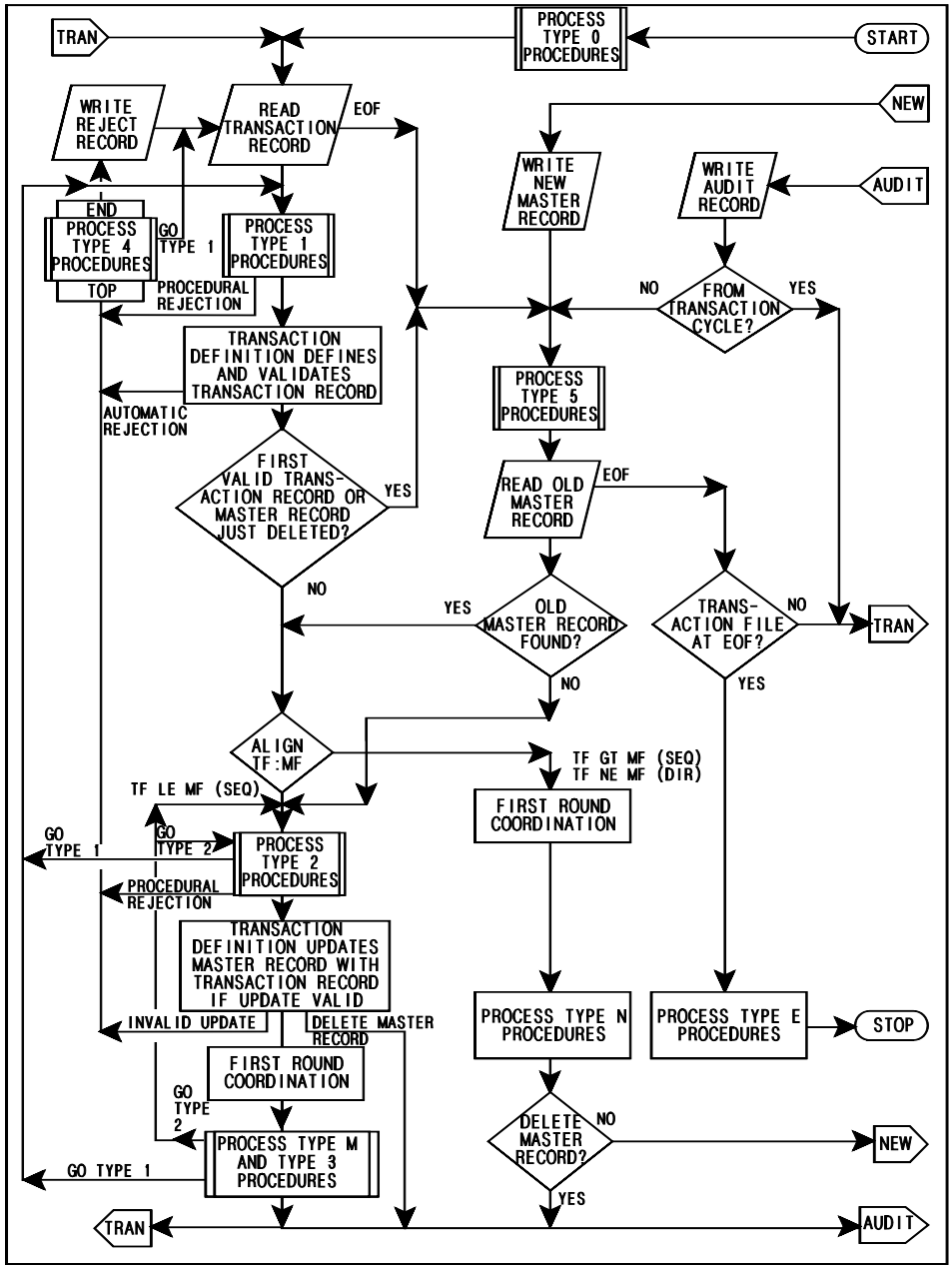



Figure 15-3 Transaction Processing Flow Including Transaction Requests and Coordinated Files

# Transaction Procedures

A transaction procedure or request consists of the same components described in Chapter 5, *The Application Cycle* and Chapter 11, *Requests*, namely, a procedure definition and up to nine report definitions. They function in the same manner with the top of procedure, output of procedure, and end of procedure branch locations. Transaction procedures, however, are enhanced with procedure type branching for transaction record retry, adding performance to transaction processing.

Transaction procedures can reference the various work areas as shown in Figure 15-2 before, during, and immediately after the master record is updated by the transaction definition specifications using the transaction record data. Transaction procedures procedurally validate, edit, and manipulate transaction records to increase and enhance the performance of the transaction cycle. In this way, the format of the transaction records are specific to the updating functions of the application, through the facility of the transaction file definition and the transaction procedures. Transaction procedures increase the independence of file maintenance from file structure.

The following transaction procedure types are available during the transaction process. See Figure 15-3.

| | |
|---|---|
| Initialization (Type 0) | The Initialization (Type 0) procedure is invoked before any file processing has begun. At this point, user code can only access storage related to temporary fields, working storage, direct-index coordinated files, and user-read coordinated files. Use this request/procedure type to read special parameter information, populate host variables prior to the first access of the Master file, and so on. Master file, transaction file, and system-read sequential coordinated file buffers may not be referenced in Type 0 requests/procedures. |
| Transaction Record Initial Validation (Type 1) | The Transaction Record Initial Validation (Type 1) procedure references the work areas immediately after the transaction record is read but before it is identified by a transaction definition. |
| Transaction File and Master File Aligned (Type 2) | The Transaction File and Master File Aligned (Type 2) procedure references the work areas after the transaction record is identified and aligned with the master record by the transaction definition but before the master record is updated. |

| | |
|---|---|
| After Update, Transaction View (Type M) | The After Update, Transaction View (Type M) procedure references the work areas immediately after the master record is updated. When type M procedures reference master record segment types that have been updated by the current transaction, VISION:Builder does not loop; it processes the occurrence of each updated segment type. Segment types not updated by the current transaction are processed with looping. |
| After Update, Entire Record View (Type 3) | The After Update, Entire Record View (Type 3) procedure references the work areas immediately after type M procedures. But when type 3 procedures reference updated segments, VISION:Builder does loop. |
| Transaction Error Handling (Type 4) | The Transaction Error Handling (Type 4) procedure references the work areas only when a transaction record is automatically or procedurally directed for rejection. |
| Before Master File Read (Type 5) | The Before Master File Read (Type 5) procedure references the work area just before a Master file record is read. At this point, user code can only access storage related to temporary fields, working storage, direct-index coordinated files, user-read coordinated files, and transaction files. Use this request/procedure type to populate host variables and perform other appropriate actions before a read of the next Master file record is attempted. For example, a Type 5 request/procedure is usually required to populate the host variables in a DB2 WHERE clause when updating tables with more than nine keys using direct read. Master file and system-read sequential coordinated file buffers may not be referenced in Type 5 requests/procedures. |

Each transaction procedure can also call subroutines procedures.

The timing and relationship of events and transaction procedures, to the transaction cycle, is best understood when presented in flowcharts. In Figure 15-4 through Figure 15-8 we expand the transaction processing flow shown in Figure 15-3.

The transaction file and master file records flows are diagramed in Figure 15-4. See Chapter 10, *Transactions* for a description.

The details and decisions of the transaction processing cycle, in relationship to the transaction procedure Types 1, 2, M and 3, and 4 are diagramed in Figure 15-5 through Figure 15-8.

In the section following the flowcharts, the transaction procedure types are discussed. There is information regarding the point within the transaction cycle that a transaction procedure is given control for execution. Also explained are the

work areas available to each transaction procedure type at their time within the transaction cycle. The information includes the facilities that are available for controlling the transaction processing flow.

In Using Transaction Procedures for an Updating Application on page 15-17, an update application example is presented in detail to demonstration how transaction procedures and the cycle can be utilized.

**Note:** The following is a guide for the On-page and Off-page connectors in Figure 15-4.

**For TRAN, see from Figure 15-4 on page 15-8 and from Figure 15-7 on page 15-11.**

**For REJECT, from Figure 15-8 on page 15-12.**

**For TYPE 1, to Figure 15-5 on page 15-9.**

**For OLD, from Figure 15-5 on page 15-9.**

**For ALIGN, to Figure 15-6 on page 15-10.**

**For TYPE 2 UNMATCH, to Figure 15-6 on page 15-10.**

**For TYPE N, from Figure 15-6 on page 15-10.**

**For NEW, to and from Figure 15-4 on page 15-8.**

**For AUDIT, from Figure 15-6 on page 15-10 and from Figure 15-7 on page 15-11 to Figure 15-4 on page 15-8.**

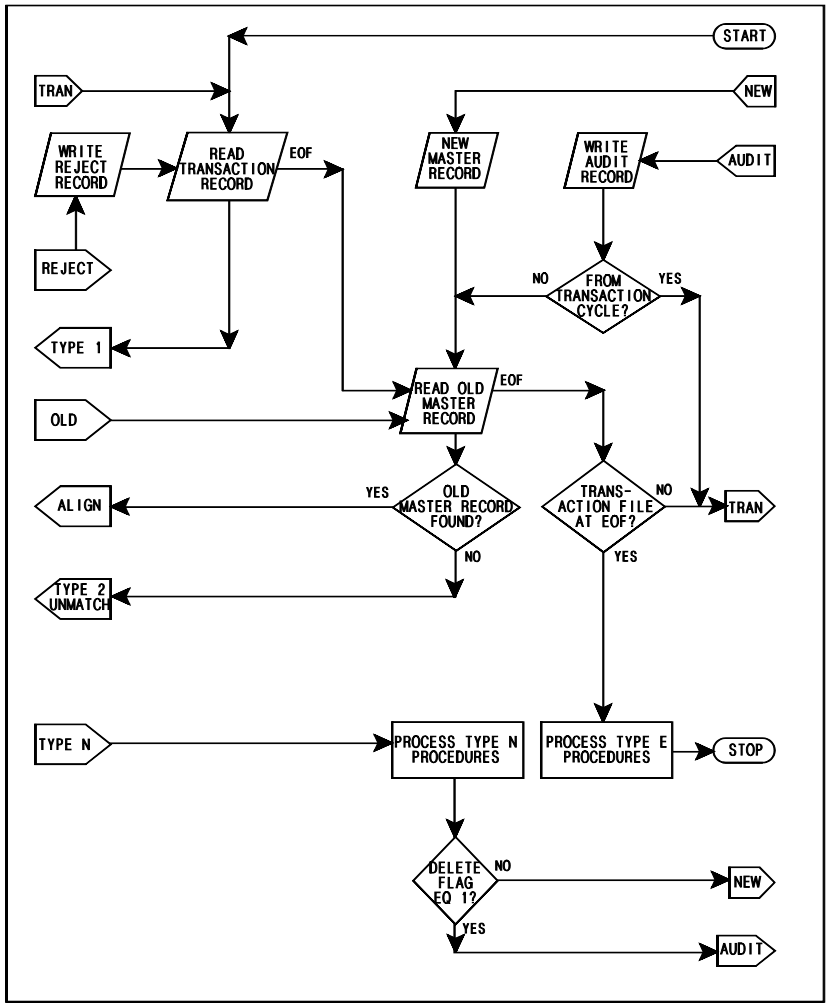



Figure 15-4 The Primary Files in Transaction Processing Flow

**Note:** The following is a guide for the On-page and Off-page connectors in Figure 15-5.

**For TYPE 1, from Figure 15-6 on page 15-10, and Figure 15-7 on page 15-11 and Figure 15-8 on page 15-12.**

**For TYPE 4, to Figure 15-8 on page 15-12.**

**For OLD, to Figure 15-4 on page 15-8.**

**For ALIGN, to Figure 15-6 on page 15-10.**

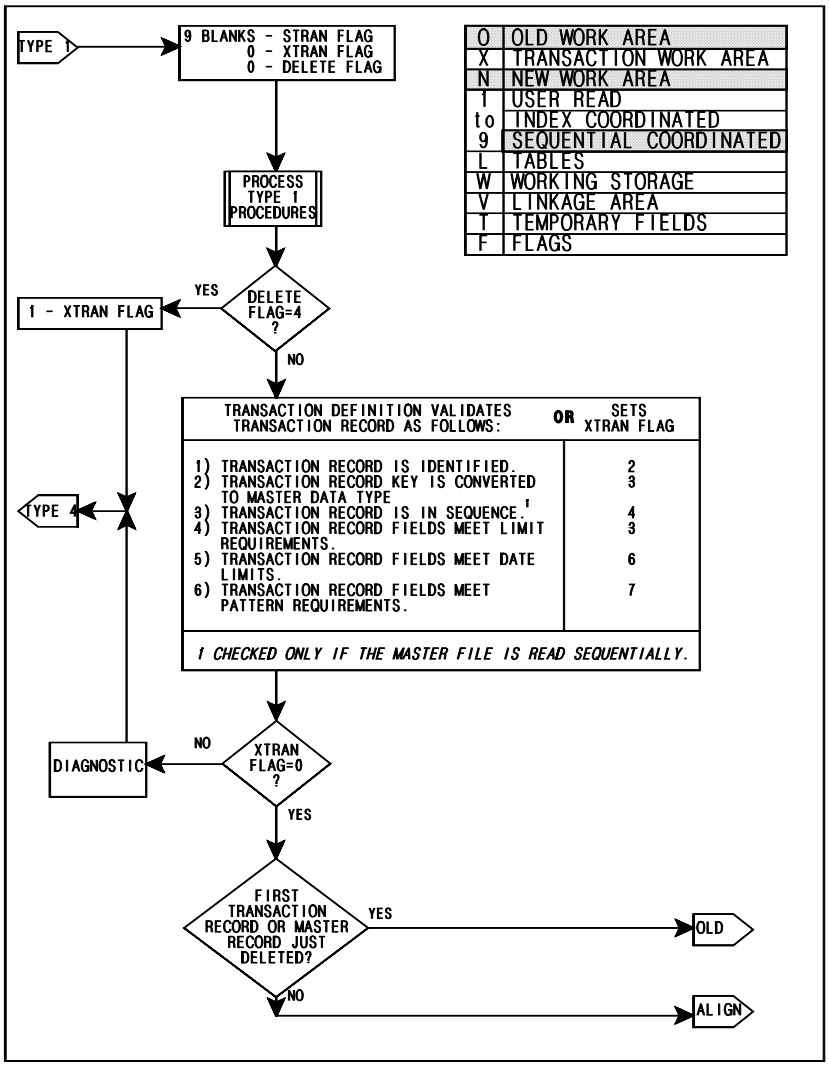



Figure 15-5 Type 1 Procedure Processing and Transaction Record Identification and Validation Flow

**Note:** The following is a guide for the On-page and Off-page connectors in Figure 15-6.

**For ALIGN, from Figure 15-4 on page 15-8 and from Figure 15-5 on page 15-9.**

**For TYPE 2 UNMATCH, from Figure 15-4 on page 15-8.**

**For TYPE 1, to Figure 15-5 on page 15-9.**

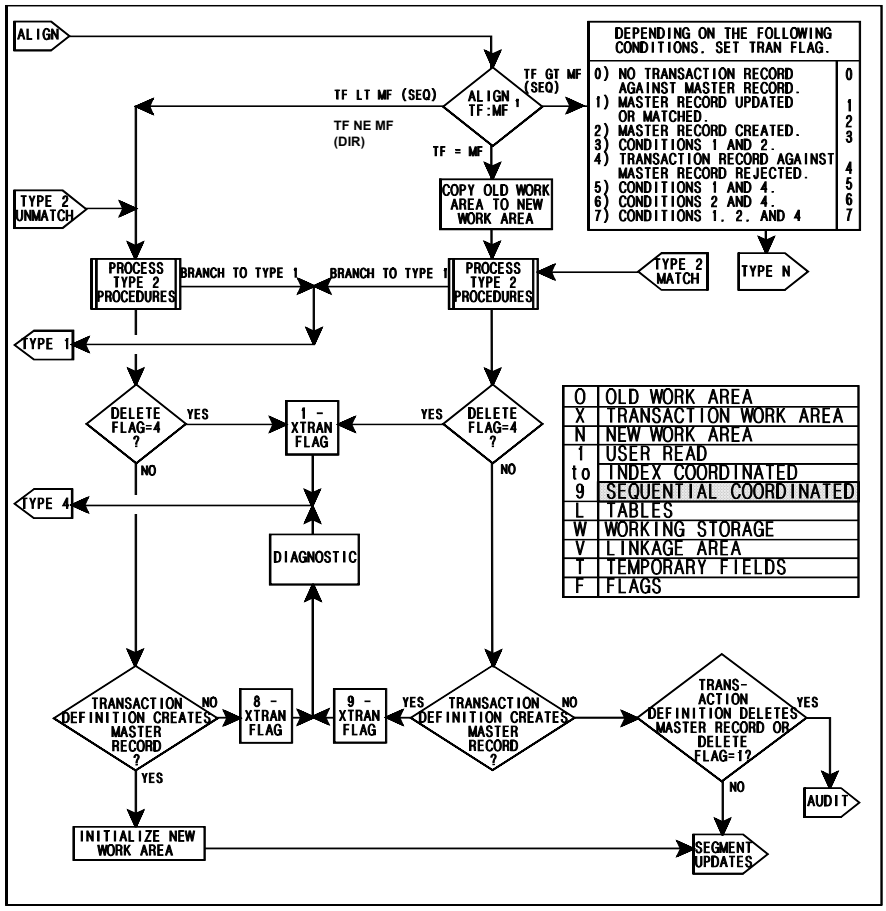
**For TYPE 4, to Figure 15-8 on page 15-12.**

**For TYPE 2 MATCH, from Figure 15-7 on page 15-11.**

**For TYPE N, to Figure 15-4 on page 15-8.**

**For AUDIT, to Figure 15-4 on page 15-8.**

**For SEGMENT UPDATES to Figure 15-7 on page 15-11**




**Note:** TF is the transaction record key field value. MF is the master record key value.

**Note:** If TF equals MF and the previous transaction record created a master record, the old work area is unavailable but the new work area is available. If TF does not equal MF, neither master work area is available.

Figure 15-6 Alignment and Type 2 Procedure Processing Flow

**Note:** The following is a guide for the On-page and Off-page connectors in Figure 15-7.

**For SEGMENT UPDATES, from** Figure 15-6 on page 15-10.

**For TYPE 4, to** Figure 15-8 on page 15-12.

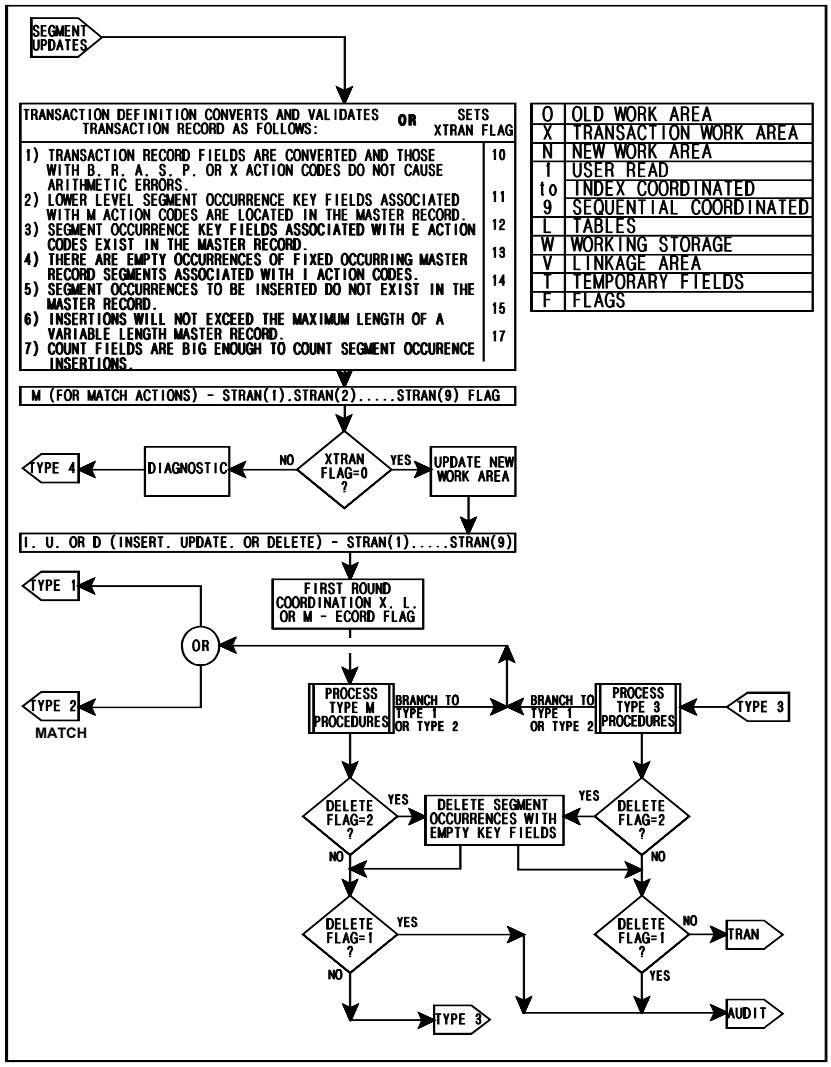
**For TYPE 1, to** Figure 15-5 on page 15-9.

**For TYPE 2 MATCH, to** Figure 15-6 on page 15-10.

**For TYPE 3, to and from** Figure 15-7 on page 15-11.

**For TRAN, to** Figure 15-4 on page 15-8.

**For AUDIT, to** Figure 15-4 on page 15-8.




**Note:** There are no automatic loops on updated segments while type M procedures are processed.

**Note:** Old work area is not available if master record was just created.

**Note:** FIND (FS) functions or RELEASE (RS) commands set the LSTATUS flag to TRAN if the occurrence referenced in the type M procedure was matched or inserted. FIND functions or RELEASE commands set the LSTATUS flag to TDEL if the occurrence referenced in type M procedure was deleted.

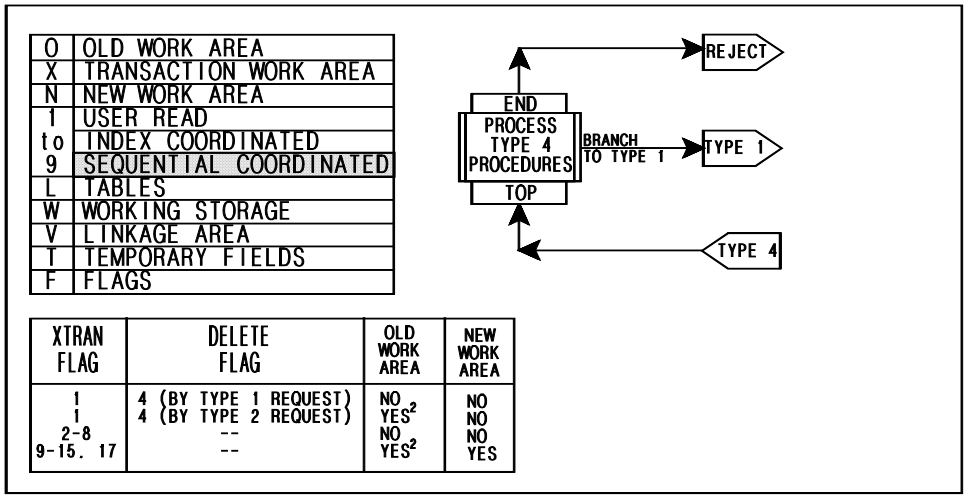
Figure 15-7 Updating Type M Procedure and Type 3 Procedure Processing Flow

**Note:** The following is a guide for the On-page and Off-page connectors in Figure 15-8.

**For REJECT, to** Figure 15-4 on page 15-8.

**For TYPE 1, to** Figure 15-5 on page 15-9.

**For TYPE 4, from** Figure 15-5 on page 15-9, Figure 15-6 on page 15-10, **and** Figure 15-7 on page 15-11.




**Note:** Old and new work areas are available to type 4 procedures according to the XTRAN/DELETE flag table shown in the figure.

**Note:** The value of YES in the OLD WORK AREA column of the table shown in the figure is only applicable if the master file was not created.

Figure 15-8 Type 4 Procedure Processing Flow

## Type 1 Procedure Processing / Transaction Record Identification/Validation

Type 1 procedures reference the work areas after the transaction record is read and the STRAN, XTRAN, and DELETE flags are reset (see Figure 15-4 on page 15-8 and Figure 15-5 on page 15-9). Type 1 procedures cannot reference the old, new, or sequential coordinated work areas. They can reference the transaction work area before it is identified, validated, or aligned by a transaction definition. In fact, type 1 procedures can direct a transaction record for rejection even before it is identified.

Type 1 procedures can be used to procedurally validate transaction records. If the transaction record is inappropriate or erroneous, a 4 is moved to the DELETE flag. When the DELETE flag equals 4 after all type 1 procedures are processed,

VISION:Builder puts a one in the XTRAN flag to indicate a procedure-directed rejection and directs control to the point in the cycle where the type 4 procedures are executed.

Type 1 procedures can translate transaction records from a legible format suited for input to an organized format more suited to the transaction definitions that specify how the data will maintain the master file. In this function, type 1 procedures can use table and indexed or user-read coordinated files. This translation includes simple error correction and even modifications to the transaction record identification field.

Transaction records can be reapplied following the type 1 procedures because type 2, type M, type 3, and type 4 procedures can branch back to type 1 procedures to retry the transaction record. This is demonstrated by the application described in Using Transaction Procedures for an Updating Application on page 15-17.

In all, type 1 procedures are the first application procedures able to reference the transaction work area. They are useful for transaction record procedural validation, procedural translation and reprocessing of transaction records in conjunction with back branches to type 1 procedures. Because they can modify the transaction work area's identification field, they improve the performance of your application's transaction records.

If the DELETE flag does not equal 4 following the type 1 procedure, the available transaction definitions are used to identify the transaction record or set the XTRAN flag to 2. The identifying transaction definition converts the transaction record key field to the master record key field data type or sets the XTRAN flag to 5. If the master file is read sequentially, the sequence of the transaction record is checked. Finally, the transaction definition performs automatic validations such as limit, date, and pattern validations.

If the transaction record does not meet any of these validations, VISION:Builder sets the XTRAN flag according to Figure 15-5, issues a diagnostic, and passes control to the type 4 procedure. Refer to the *VISION:Builder ASL Reference Guide* or the *VISION:Builder Reference Summary* for a complete list of the XTRAN flag settings.

If the transaction record is valid, VISION:Builder reads the old master record if this is the first transaction record or if a master record was just deleted; otherwise, VISION:Builder proceeds to align the transaction record with the current master record.

## Alignment and Type 2 Procedure Processing

Type 2 procedures can reference all but sequentially coordinated work areas. Because they can do this before the master work area is updated based on the transaction definition specifications, they can most appropriately determine the consequences of the pending update. With this information, type 2 procedures are best suited to customize the transaction process based on its actual updates.

In type 2 procedures, the transaction record and master record have been aligned after the transaction record is identified and validated by a transaction definition. There now exists a current master record in the work areas (see Figure 15-6 on page 15-10).

If the master file is directly read, alignment occurs only if a master record was found with the proper key field value. Type 2 procedures can be executed without a matching master record. In Figure 15-6, see the flow at the TYPE 2 UNMATCH label on the upper left of the figure.

If the alignment test determines that the transaction file record key is lower than a sequentially read master file record key, the Type 2 procedures are executed without a matching master file record. In Figure 15-6, see the flow at the Type 2 UNMATCHED label.

If the alignment test determines that the transaction file record key is greater than a sequentially read master file record key, the transaction process is finished for the master file record. The TRAN flag is set according to the box on Figure 15-6 and processing of the master file record continues with the normal (Type N) procedures.

If the alignment test determines that the transaction file record key matches the master file record key, the Type 2 procedures are executed with a matching master file record.

When the alignment process results in a match, the old and new work areas are available. When the alignment process results in a non-match, the old work area is not available. The new work area may be available if a previous transaction record has created a new master record. The TRAN flag can be tested to determine if the new work area is available.

Type 2 procedures can back branch to the type 1 point within the transaction cycle to retry the transaction record. In this case, the type 1 back branch is taken to the type 1 procedures in Figure 15-5. Because type 2 procedures can reference the master and transaction work areas, they can branch to type 1 to better identify the transaction record or to improve the alignment. Also, because the transaction and master work areas can be referenced, type 2 procedures can anticipate the update and modify the transaction record to optimize the update.

Type 2 procedures can also direct the transaction record for rejection by placing a 4 in the DELETE flag. They can do this as they reference the master and transaction work areas. Type 2 procedures determine the validity of the transaction record and its prescribed update. If the DELETE flag is not set to 4 after all type 2 procedures are executed, the master record is updated based on the transaction definition specifications.

When processing is in the UNMATCH leg of the flowchart in Figure 15-6, the transaction definition must create a master record; otherwise, VISION:Builder sets the XTRAN flag to 8, issues a diagnostic, and proceeds to the type 4 procedures. This is because the UNMATCH leg assumes no current master record. If a master record is created, VISION:Builder initializes the new work area and proceeds to update it with the transaction data.

When processing is in the MATCH leg of the flowchart, the transaction definition must not create a master record because one is matched. If it attempts to create a master record, VISION:Builder sets the XTRAN flag to 9, issues a diagnostic, and proceeds to the type 4 procedures. In this leg, the transaction definition can delete the master record. In this case, the master record can be written to the audit file and another transaction can be read (see Figure 15-4 on page 15-8). VISION:Builder proceeds to update the master record if it is not deleted.

## Updating and Type M and Type 3 Procedure Processing Flow

After alignment and type 2 procedure processing and, if the master record is not deleted, the transaction definition updates the master record segments. However, before this occurs, the segment updates are aligned in the master record and the associated transaction definition action codes are validated (see Figure 15-7).

In the upper left box (see Figure 15-7), VISION:Builder sets the XTRAN flag if the segment level alignments or update actions are inappropriate. The STRAN flag is set according to the segment matches. If the XTRAN flag is set, VISION:Builder issues a diagnostic and transfers control to the type 4 procedures; otherwise, VISION:Builder updates the new work area according to the transaction definition and the current transaction record. Immediately after updating the new work area, VISION:Builder sets the STRAN flag to indicate insert, update, or delete actions for future reference. Finally, VISION:Builder performs first round coordination between the sequential coordinated files and the master record and sets the ECORD flag accordingly.

The type M procedures are executed. They can reference all work areas except the old work area if a master record was just created. Unlike any other procedure types, type M procedures do not initiate loops on segments that have been updated. Likewise, when type M procedures attempt to find or release segment occurrences just updated using FIND (FS) functions or RELEASE (RS) commands, VISION:Builder suppresses the operation and puts TRAN in the LSTATUS flag if the occurrence was matched or puts TDEL in the LSTATUS flag if the occurrence was deleted.

In effect, VISION:Builder holds the updated path in the master record so that type M procedures can easily concentrate on the updated portion of the master record. Type M procedures are useful to report updating activity. Other post-updating tasks are best handled with type 3 procedures that are described next.

The STRAN flag is very useful to type M procedures because it conveys the updates at all segment levels. The STRAN flag can be interrogated so that type M procedures can quickly pinpoint master record updates for reporting and looping.

Type M procedures can branch back to type 1 or 2 procedures to retry the transaction record. They can do this with knowledge of the update. This section, however, reserves the discussion of transaction retry after updating until type 3 procedures are described.

Type M procedures cannot direct a transaction record for rejection because it has already updated the new work area. Type M procedures can reference the transaction work area and can include transaction record fields in their updating report.

Type M procedures can direct a segment occurrence for deletion by emptying its key field, and placing a 2 in the DELETE flag. If the DELETE flag equals 2 after all type M procedures are processed, all segment occurrences having empty key fields are deleted.

Type M procedures can also direct an entire master record for deletion by placing 1 in the DELETE flag. If the DELETE flag equals 1 after all type M procedures are processed, VISION:Builder deletes the master record, optionally writes it to the audit file, and reads another old master record. If the DELETE flag does not equal 1 after all type M procedures are processed, control passes to the type 3 procedures.

Type 3 procedures are similar to type M procedures with three exceptions:

- VISION:Builder automatically loops on all referenced lower level segments.
- FIND (FS) functions and RELEASE (RS) commands do not affect the LSTATUS flag and are implemented as usual.
- If the master record is not deleted after all type 3 procedures are processed, the next transaction record is read.

Type 3 procedures can increase the performance of transaction records because they can reference and modify the transaction work area and branch back to type 1 or 2. Type 3 procedures can procedurally reformat the transaction work area including its identification and branch back to type 1. In this way, a different transaction definition can identify the transaction record and different update action codes can be employed. Also, the contents of the transaction record can be reformatted to update different segments or even different paths in the master record. Type 3 procedures can process a single transaction record as several transaction records against the master file.

Using Transaction Procedures for an Updating Application on page 15-17 demonstrates these procedures. Your application can streamline user maintenance and reduce redundant tasks. This is because the transaction record can be processed as a structured record even though it is not defined as such.

Type 3 procedures are also useful for reporting batch and control totals showing the cumulative effect of transaction records on the master file.

## Transaction Record Rejection and Type 4 Procedure Processing Flow

Type 4 procedures are processed only after the transaction record is directed for rejection by VISION:Builder, a type 1 procedure, or a type 2 procedure (see Figure 15-8). VISION:Builder directs transactions for rejection when it cannot process them according to the identifying transaction definition.

A type 1 or 2 procedure directs a transaction record for rejection by placing a 4 in the DELETE flag (VISION:Builder puts a 1 in the XTRAN flag). Type 4 procedures, however, can always reference the work areas before a transaction record is actually rejected and can reject it as directed or modify and resubmit it with a branch back to type 1 procedures.

Type 4 procedures can provide diagnostic reports of the transaction record, showing its shortcomings and the work areas at the time of its rejection. Flags (especially the XTRAN and STRAN flags), temporary fields, transaction work area, and master work areas (if the transaction record is not directed for rejection by a type 1 procedure) can be referenced for this purpose.

Reports produced from type 4 procedures can provide information needed to correct transaction records. After all type 4 procedures are processed and, if none has branched to the type 1 procedures, the transaction record is rejected, optionally written to the reject file, and another record is read from the transaction file.

If a type 4 procedure elects to retry the transaction record, it can branch back to the type 1 point within the transaction cycle. Before branching, however, type 4 procedures should repair the transaction record procedurally using flags, temporary fields, transaction work area, and master work areas if possible. Type 4 procedures can modify the transaction record so that it might pass validation. Type 4 procedures can also report procedural repairs.

## Using Transaction Procedures for an Updating Application

Normally, transaction definitions are sufficient to maintain application files. However, to refine the updating process for specific application validation or data entry or to adapt application updating to process transaction records of variable format, transaction procedures are appropriate.

This section uses an updating application example to demonstrate the use of transaction procedures. We have expanded on the basic update process discussed in Chapter 10, *Transactions*. In the expanded updating example we have made a change to the ITEM transactions records and added several reports to provide more information about the customer master update process.

The 4ADDITEM transaction record shown in Chapter 10, *Transactions* was designed to add one item to the customer master and included the item name and price. We have designed a new transaction record type identified as the 5ADDITEM record. This is included in the ORDENTRY transaction group definition. The 5ADDITEM transaction record is designed to add from one to five items to the customer master per input transaction record. The item name and price are not included, only the item number and quantity values are provided. There are five sets of item and quantity data values. This will reduce the effort

needed to prepare the item transaction records. Figure 15-9 shows the 5ADDITEM record format. You can compare this new format with the 4ADDITEM format shown in Figure 10-5 on page 10-9 to see the differences..

| TRAN ID (5ADDITEM) | CUST. NO. | ORDER NO. | ITEM 1 | | ITEM 2 | | ITEM 3 | | ITEM 4 | | ITEM 5 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | NO. | QTY. NO. | NO. | QTY. NO. | NO. | QTY. NO. | NO. | QTY. NO. | NO. | QTY. NO. |
| 1 8 | 9 13 | 14 18 | 19 25 | 26 30 | 31 37 | 38 42 | 43 49 | 50 54 | 55 61 | 62 66 | 67 73 | 74 78 |

Figure 15-9 New 5ADDITEM Transaction Record Format

The ORDENTRY transaction group definition includes the new 5ADDITEM transaction and the TRANWORK file definition includes references for the data fields. These are utilized in an updating application. Figure 15-10 shows a portion of the TRANWORK file definition glossary listing that relates to the new 5ADDITEM records. A complete glossary list for the TRANWORK file definition is in TRANWORK File Definition on page D-145.

```
                                    DETAILED GLOSSARY BY NAME FOR                                  GLOSSARY PAGE      1
                                      FILE DEFINITION - TRANWORK
DATE CATALOGED - 02/11/02 14:27:06                                          DATE LAST UPDATED - 02/11/02 14:27:06
CATALOGED BY -                                                              LAST UPDATED BY -
EXPIRATION DATE -                                                           DATE LAST USED -
FILE IDENTIFICATION =                                                       RECORD FORMAT = FIXED BLOCKED
NUMBER OF SEGMENTS IN FILE = 1                                              RECORD SIZE = 80
NUMBER OF FIELDS IN FILE = 20                                               BLOCK SIZE = 800

                                  ********************************
                                  *    SEGMENT   1, LEVEL 1     *
                                  *   SEGMENT NAME = WORK01     *
                                  ********************************
SEGMENT OCCURS N TIMES = 1                                  KEY FIELD 1 = CUSTNO     TYPE = C   LENGTH = 5
SEGMENT SIZE = 80
NUMBER OF FIELDS IN SEGMENT = 20
-------------------------------------------------------------------------------------------------------------------------------
FIELD   FIELD   FIELD    FIELD   FIELD    DEC    CNT FIELD  EDIT CODES     EDIT    OUTPUT   LINE  DATA BASE NAME/
NAME    TYPE   LOCATION  LENGTH  RNDING  PLACES  FOR SGMT   ( ) ( ) ( )   LENGTH   WIDTH     NO   *** COLUMN HEADING ***
-------------------------------------------------------------------------------------------------------------------------------
TRANID    C        1        8                                                8        8
                                                                                          1    ***     TRANS      ***
                                                                                          2    ***      ID        ***
CUSTNO    C        9        5                                                5        5
                                                                                          1    ***      CUST      ***
                                                                                          2    ***      NO.       ***
ORDERNO   C       14        5                                                5        5
                                                                                          1    ***     ORDER      ***
                                                                                          2    ***      NO.       ***
ITM5ORD1  C       19        7                                                7        7
                                                                                          1    ***      ITEM      ***
                                                                                          2    ***     ORDR 1     ***
ITM5QTY1  Z       26        5                                                7        7
                                                                                          1    ***     ITEM       ***
                                                                                          2    ***     QTY 1      ***
ITM5ORD2  C       31        7                                                7        7
                                                                                          1    ***      ITEM      ***
                                                                                          2    ***     ORDR 2     ***
ITM5QTY2  Z       38        5                                                7        7
                                                                                          1    ***     ITEM       ***
                                                                                          2    ***     QTY 2      ***
ITM5ORD3  C       43        7                                                7        7
                                                                                          1    ***      ITEM      ***
                                                                                          2    ***     ORDR 3     ***
ITM5QTY3  Z       50        5                                                7        7
                                                                                          1    ***     ITEM       ***
                                                                                          2    ***     QTY 3      ***
ITM5ORD4  C       55        7                                                7        7
                                                                                          1    ***      ITEM      ***
                                                                                          2    ***     ORDR 4     ***
ITM5QTY4  Z       62        5                                                7        7
```

Figure 15-10 A Portion of the TRANWORK File Definition Glossary (Page 1 of 2)

```
                                                                                  1    ***    ITEM      ***
                                                                                  2    ***    QTY 4     ***
ITM5ORD5  C       67       7                                   7       7
                                                                                  1    ***     ITEM     ***
                                                                                  2    ***    ORDR 5    ***
ITM5QTY5  Z       74       5                                   7       7
                                                                                  1    ***    ITEM      ***
                                                                                  2    ***    QTY 5     ***
```

Figure 15-10 A Portion of the TRANWORK File Definition Glossary (Page 2 of 2)

Figure 15-11 shows the 5ADDITEM transaction definition within the ORDENTRY transaction group definition.

```
                                    GLOSSARY BY LEVEL NUMBER FOR                                  GLOSSARY PAGE      3
                                    TRANSACTION GROUP - ORDENTRY

                                ***************************************
                                *   3  OF  5  IDENTIFIERS IN GROUP   *
                                ***************************************
**********    ACTIONS IN IDENTIFIER = 4          IDENTIFIER ONE = (5ADD)          IDENTIFIER TWO = (ITEM)
*5ADDITEM*    DEFAULT CREATE = NO                IDENTIFIER LOCATION = 1          IDENTIFIER LOCATION = 5
**********    DEFAULT INSERT AT LEVEL 3          IDENTIFIER LENGTH = 4            IDENTIFIER LENGTH = 4
                                                 IDENTIFIED IF EQUAL              IDENTIFIED IF EQUAL
-----------------------------------------------------------------------------------------------------------------------
MASTER  ..........TRANSACTION DATA..........
FILE    FIELD     FIELD     FIELD    DECIMAL   ACTION   LEVEL          MINIMUM                  MAXIMUM
FIELD   TYPE     LOCATION   LENGTH   PLACES     CODE    NUMBER          VALUE                    VALUE
-----------------------------------------------------------------------------------------------------------------------
CUSTNO    C          9        5                M        1          EDIT = Y9999
ORDERNO   C         14        5                M        2
ITEMORD   C         19        7                M        3
ITMQTYOR  Z         26        5                R        3
```

Figure 15-11 5ADDITEM Transaction Definition Glossary Listing

The update application also used the 3ADDORDR Transaction from the ORDENTRY transaction definition. Descriptions of the 3ADDORDR and 5ADDITEM transaction definitions are described in The Update Application Flow on page 15-23.

You will notice that transaction record fields are rearranged, omitted, or combined. The order-entry department uses the new format to save time and effort in adding items and to increase the reliability and legibility of its item transaction records. The order-entry update application incorporates transaction procedures to procedurally manipulate and apply one transaction record as several transaction records to the customer file, each time updating different segment occurrences.

## Elements of the Updating Run

Figure 15-12 shows the Update Application Run Control Commands we are using in our example for this chapter.

```
CONTROL NAME UPDCUSTM TERM LISTGEN
FILE MASTER INPUT  CUSTMAST
FILE MASTER OUTPUT
FILE TRAN   TRANWORK GROUPS ORDENTRY
FILE CORD1  ITEMMAST DIRECT BY X.ITM5ORD1
FILE REPORT
```

Figure 15-12 Update Application Run Control Commands

The source listing of the run control commands is in Run Control Commands on page D-161.

We have specified the Master File as INPUT and OUTPUT. This means that we will be creating a separate updated version of the Customer Master File with our application. The CUSTMAST file definition describes the master file records in the old and new work areas.

The Transaction File records are described by the TRANWORK file definition. The updating actions to be performed using the transaction file records are described in the ORDENTRY transaction group definition. The transaction work area is referenced using the X qualifier.

In the updating application, a coordinated file will be used. The ITEMMAST file definition describes the Cord1 File. This file will be read directly using a data field from the transaction file work area, X.ITM5ORD1. We will be getting the Item Name and Price from this file.

We specified a Report File since we will be producing several reports detailing the update run results.

The complete glossary listing for the files in this update application run are in Chapter D, *Complete Sample Applications*, starting at Figure D-62.

Figure 15-13 shows a list and brief description of the Update Application Procedures. The procedure name, type, and description provide an overview of the processing that will be performed by the update application. The complete details of the update application are explained in The Update Application Flow on page 15-23.

| Procedure Name | Type | Description |
|---|---|---|
| TRANPREP | 1 | Produce a report of the transaction records.<br>Edit Item quantity values and reject transaction as needed. |
| ORDERCHK | 2 | Check transaction order dates versus the master file.<br>Report all transactions that pre-date the master file dates. |
| UPDATRPT | M | Put the Item Name and Price into the customer master.<br>Delete the Item segment if no Item name or price is found.<br>Report updated customer record information. |
| ITEMLOOP | 3 | Manipulate Item transaction records.<br>Check and process the multiple sets (up to 5 sets) of Item and Quantity data. (This is done in a subroutine procedure.)<br>Re-process (Retry) the Item transaction after manipulating the item number and quantity data. |
| TRANERR | 4 | Process and Report Rejected Transactions.<br>Reports are in a Subroutine Procedure.<br>Manipulate the rejected Item transaction records.<br>Check and process the multiple sets (up to 5 sets) of Item and Quantity data. (This is done in a subroutine procedure.)<br>Re-process (Retry) the Item transaction after manipulating the item number and quantity data. |
| REJRPTS | Sub | Produce the Rejected Transaction Reports.<br>A report for the Order transactions.<br>A report for the Item transactions.<br>A report for the Unknown transactions. |
| ITEMCHKR | Sub | Check and process the multiple sets (up to 5 sets) of Itemand Quantity data. |
| MAIN | Normal | Report all the Customer Master Records. |

Figure 15-13 The Update Application Procedures

The flow diagram of the update application is shown in Figure 15-14. Review this figure while you read the following description of the update application. Figure 15-14 is a condensed and specialized version of the flow diagrams in Transaction Procedures on page 15-6.

**Note:**

1. **TF is the transaction record key field value. MF is the master record key value.**
2. **Branch to type 1.**
3. **DELETE flag equals 4.**
4. **Invalid update. Transaction rejected.**
5. **CALL to sub-procedure ITEMCHKR**
6. **CALL to sub-procedure REJRPTS**

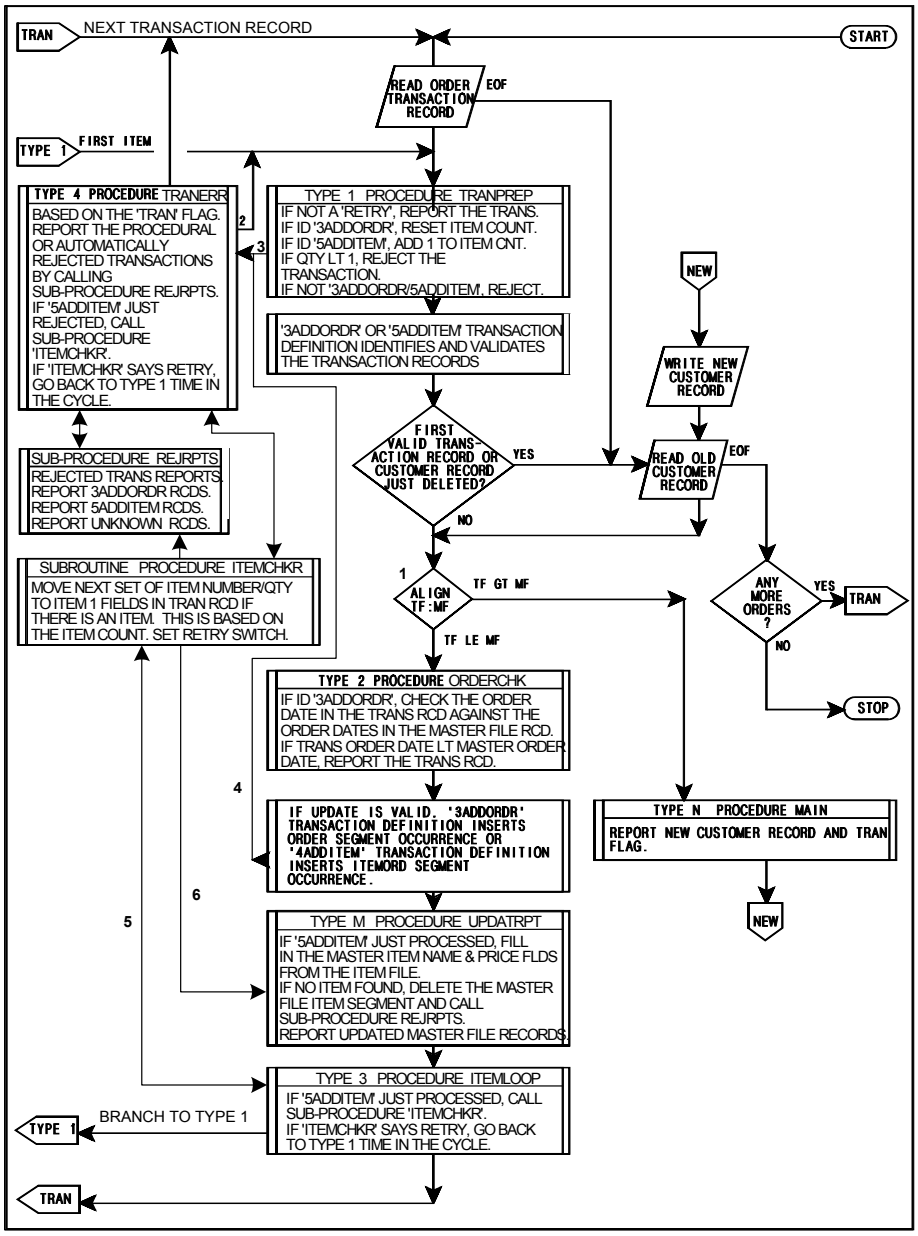



Figure 15-14 The Order-Entry Application Flow

# The Update Application Flow

After each transaction record is read, it is Type 1 time. Any Type 1 procedures are executed. In the update application, procedure TRANPREP is executed. See Figure 15-15.

```
TRANPREP: PROC TYPE1
;
; This TYPE 1 Procedure is executed for each new transaction record
;
;   Either Read from the M4TRAN Input file or
;   or because of a RETRY, a TRANSFER TYPE_1 Command
;
ITMCOUNT: FIELD P 1  HEADING 'COUNT' 'ITEM TRANS'
WORK1C:   FIELD C 1  HEADING 'WORK' 'CHAR 1'
ERRNOTE:  FIELD C 25 HEADING 'ERROR' 'COMMENT'
RETRYSW:  FIELD C 3  HEADING 'RETRY' 'SWITCH' INIT 'NO '
;
LET T.ERRNOTE = ' '
IF T.RETRYSW = 'NO'                 ; RETRY ? Entry
  REPORT F.M4TRAN X.ALL80           ; No, list Tran Rcd
  TITLE 'LISTING OF TRANSACTION RECORDS 80/80 WITH RECORD COUNT'
  END REPORT
END IF
IF X.TRANID = '3ADDORDR'            ; if a 3ADDORDR Tran Rcd
  LET T.ITMCOUNT = 0                ; reset the ITEM COUNTER
END IF
IF X.TRANID = '5ADDITEM'            ; if a 5ADDITEM Tran Rcd
  LET T.ITMCOUNT = T.ITMCOUNT + 1   ; Add to the ITEM COUNTER
  IF X.ITM5QTY1 LT 1                ; Is there a Quantity
   LET F.DELETE = 4                 ; No, reject the transaction
   LET T.WORK1C = T.ITMCOUNT        ; Convert item count
   COMBINE 'QTY INVALID FROM SET' T.WORK1C STORE T.ERRNOTE BLANKS 1
  END IF
END IF
IF X.TRANID NE '3ADDORDR' '5ADDITEM'     ;Expected IDs?
   LET F.DELETE = 4                      ; No, reject the tran rcd
  END IF
;
END PROC ;TRANPREP
```

Figure 15-15 Type 1 Procedure TRANPREP

The source listing of the TRANPREP Procedure is in Type 1 Procedure TRANPREP on page D-162.

In the TRANPREP we will do some preparation work and initial editing. One thing to keep in mind, this procedure will be entered every time a transaction record is read. It will also be entered when other procedures transfer control back to Type 1 time. In our update application, procedures ITEMLOOP and TRANERR branch back to the Type 1 point within the transaction cycle for the processing of the multiple items in the 5ADDITEM transaction records. This is discussed later.

When we have entered the TRANPREP procedure from a transaction record read, the RETRYSW is 'NO'. In this case, we want to list the new transaction record in a report for reference.

When a 3ADDORDR transaction is encountered, we reset the ITMCOUNT to zero. The ITMCOUNT is used to count the multiple items processed within an order from the 5ADDITEM transaction records.

When a 5ADDITEM transaction is encountered, we increment the ITMCOUNT by 1. We will count the initially read 5ADDITEM transaction record along with each retry of the same transaction record. We also check the ITM5QTY1 field in the transaction record to see the quantity is less than 1. We will reject the 5ADDITEM transaction record if we do not have a positive quantity by setting the DELETE flag to 4.

We also reject any transaction record that does not contain '3ADDORDR' or '5ADDITEM' in the TRANID field of the transaction record by setting the DELETE flag to 4. We know that the ORDENTRY transaction group contains several other transaction identifiers that do not apply to our update application. Our update application is only designed for handling the 3ADDORDR and 5ADDITEM transaction record types.

When all Type 1 procedures are completed, VISION:Builder rejects the current transaction record whenever the DELETE flag has a setting of 4. This will direct the transaction processing cycle to Type 4 time. Type 4 time is discussed later in this section.

Following the Type 1 procedures, the transactions are given to the automatic part of the transaction cycle for validation. The ORDENTRY transaction definition is used identify and validate the transaction records. Figure 15-16 shows the 3ADDORDR and 5ADDITEM identifiers from the ORDENTRY transaction definition.

```
**********   ACTIONS IN IDENTIFIER = 8          IDENTIFIER ONE = (3ADD)          IDENTIFIER TWO = (ORDR)
*3ADDORDR*   DEFAULT CREATE = NO                IDENTIFIER LOCATION = 1          IDENTIFIER LOCATION = 5
**********   DEFAULT INSERT = NO                IDENTIFIER LENGTH = 4            IDENTIFIER LENGTH = 4
                                                IDENTIFIED IF EQUAL              IDENTIFIED IF EQUAL
-------------------------------------------------------------------------------------------------------------------------
MASTER  ..........TRANSACTION DATA..........
FILE    FIELD     FIELD     FIELD    DECIMAL   ACTION   LEVEL          MINIMUM                  MAXIMUM
FIELD   TYPE     LOCATION   LENGTH   PLACES     CODE    NUMBER          VALUE                    VALUE
-------------------------------------------------------------------------------------------------------------------------
CUSTNO    C          9        5                M        1          EDIT = Y9999
ORDERNO   C         14        5                I        2
ORDRDATE  C         19        6                R        2                  DATE VALIDATION
ORPERSON  C         25        3                R        2          EDIT = -9-9-9
ORDPONUM  C         28        5                R        2
ORDDUDAT  C         33        6                R        2                  DATE VALIDATION
ORDINVGN  C         39        1                R        2          EDIT = Z
ORDCMPLT  C         40        1                R        2


**********   ACTIONS IN IDENTIFIER = 4          IDENTIFIER ONE = (5ADD)          IDENTIFIER TWO = (ITEM)
*5ADDITEM*   DEFAULT CREATE = NO                IDENTIFIER LOCATION = 1          IDENTIFIER LOCATION = 5
**********   DEFAULT INSERT AT LEVEL 3          IDENTIFIER LENGTH = 4            IDENTIFIER LENGTH = 4
                                                IDENTIFIED IF EQUAL              IDENTIFIED IF EQUAL
-------------------------------------------------------------------------------------------------------------------------
MASTER  ..........TRANSACTION DATA..........
FILE    FIELD     FIELD     FIELD    DECIMAL   ACTION   LEVEL          MINIMUM                  MAXIMUM
FIELD   TYPE     LOCATION   LENGTH   PLACES     CODE    NUMBER          VALUE                    VALUE
-------------------------------------------------------------------------------------------------------------------------
CUSTNO    C          9        5                M        1          EDIT = Y9999
ORDERNO   C         14        5                M        2
ITEMORD   C         19        7                M        3
ITMQTYOR  Z         26        5                R        3
```

Figure 15-16 The 3ADDORDR and 5ADDITEM Transaction Identifiers

The 3ADDORDR transaction records match an existing customer record (Action M on CUSTNO) and will insert a new order segment (Action I on ORDERNO). The remaining data in the transaction record is replaced (Action R) into the appropriate customer master field when data is present. The 3ADDORDR transaction record will be automatically rejected under the following conditions. If it does not match

a customer record; if it matches an existing order; if it contains invalid order or due date values; or it contains invalid values for the customer number, order person, or order invoice generated. For more details, see the description of Figure 10-8 on page 10-12.

The 5ADDITEM transaction records match an existing customer record, an existing order, and an item number value (Action M on CUSTNO, ORDERNO, and ITEMORD). The quantity ordered is replaced, if present, into the customer record (Action R on ITMQTYOR). The transaction record will be automatically rejected if it does not match a customer record and an existing order. If a matched item number value is not found, an item segment will automatically be inserted (Default Insert at Level 3). The quantity ordered was procedurally edited in our Type 1 procedure TRANPREP.

Following this identification and validation process, the valid transaction records are aligned with the customer master file records as detailed in Figure 15-6 on page 15-10.

The alignment is the classic Low-Equal-High scenario. When the transaction record key value is lower than the master file record key, there is no matching master record and the transaction record is not aligned. When the transaction record key value is equal to the master file record key, the matching master record and the transaction record are aligned. When the transaction record key value is higher than the master file record key, the master file record has no more transactions, all matching transactions (if any) were applied to the master record. The new master record can be processed by standard (Normal) procedures outside the transaction processing cycle.

When the alignment produces the transaction record key low or equal conditions, it is Type 2 time is the transaction processing cycle. Any Type 2 procedures are executed at this time. In our update application, the Type 2 procedure ORDERCHK is executed. See Figure 15-17.

```
ORDERCHK: PROC TYPE2
;
; This TYPE 2 Procedure is executed for each new transaction record
;  not "automatically' rejected based on the Transaction Definition
;
; We are aligned with the Customer Master Record, but the update from
; this Tran Rcd has not been applied yet.
;
; For the 3ADDORDR Transaction Records
;  we will check the Tran Rcd "order date" to see if it pre-dates
;  any other orders in the customer record to provide a information
;  listing for the order desk
;
MSTRDATE: FIELD D 4 HEADING 'MASTER' 'DATE' ; Lilian Date
TRANDATE: FIELD D 4 HEADING 'TRAN'    'DATE' ; Lilian Date
;
IF X.TRANID = '3ADDORDR'
  LET T.TRANDATE = X.ORDRDATE         ; convert date to Lilian
```

Figure 15-17 Type 2 Procedure ORDERCHK (Page 1 of 2)

```
  LET T.MSTRDATE = N.ORDRDATE         ; convert date to Lilian
  IF  T.TRANDATE < T.MSTRDATE
   REPORT X.TRANID X.CUSTNO X.ORDERNO T.TRANDATE T.MSTRDATE F.M4TRAN
   TITLE LINE 'LIST OF ORDERS WITH ORDER DATES EARILER THAN MASTER',
              ' FILE ORDER DATES'
   END REPORT
  END IF
END IF
END PROC ;ORDERCHK
```

Figure 15-17 Type 2 Procedure ORDERCHK (Page 2 of 2)

The source listing of the ORDERCHK Procedure is in Type 2 Procedure ORDERCHK on page D-163.

In procedure ORDERCHK, we want to produce an information report. At Type 2 time, the transaction record and the customer master record are aligned. We can now compare data from both files and produce a report. We want a report that shows any transactions that contain order dates the pre-date any order dates currently in the matching customer record.

When the current transaction record is a 3ADDORDR record, we will check the order date. We have defined two temporary fields as date (field Type D) that will contain our order dates in Lilian Date Format. This makes the comparisons easier. VISION:Builder automatically converts dates to Lilian on replacement operations (the LET). If the transaction record order date is less than the customer record order date, produce a line on the report. Since the order date is in a lower level segment of our structured customer record, VISION:Builder will automatically loop through all order segment occurrences.

After all the Type 2 procedures are executed, and if the transaction record is valid and not procedurally deleted, the customer master record is updated based on the transaction identifier specifications. At this point, the transaction record data has been applied to the master record and any backout would need to be done procedurally.

Following the update of the customer master record, it is Type M time. The transaction record and customer master record are still aligned. We can check and supplement the update as needed. At this point, any Type M procedures are executed. Note that in Type M procedures, the segment path matched by the transaction record is locked on specific customer master record segment occurrences and no looping will take place on those segment types.

In our update application, the Type M procedure UPDATRPT is executed. See Figure 15-18.

```
UPDATRPT: PROC TYPEM
;
; This TYPE M Procedure executes after each Transaction Record
; Updates the Customer Record.
;
; References to Segments just updated by the current Transaction Record
; will not cause Looping on those Segment Types.
; If the current transaction record is an 5ADDITEM,
;  We need to get the ITEM NAME and PRICE from the ITEM File into
;  the Item Segment of the Customer Master.
;
;  A Direct Read of the ITEM File (CORD1) using the ITEM Order Id
;  If we do not find an Item Record, DELETE the segment from the
;  Customer Record and report the error
;
IF X.TRANID = '5ADDITEM'                ; Item Trans
 IF 1.ITEMNO = 1.ITEMNO                 ; Yes, This READS Item File
  LET N.ITEMNAME = 1.ITEMNAME           ; RCD Found, Get data
  LET N.ITMPRICE = 1.ITMPRICE
 ELSE                                           ; No RCD Found
  LET T.ERRNOTE = 'ITEM ORDERED NOT ON FILE'
  CALL REPORT REJRPTS                           ; List error
  LET N.ITEMORD = ' '                           ; Clear Segment Key
  LET F.DELETE = 2                              ; Mark SEG for DELETE
  GO TO ENDPROCM                                ; DO NOT REPORT
 END IF
END IF
REPORT CUSTNO ORDERNO ORDRDATE ORPERSON ORDPONUM ORDDUDAT,
       ORDINVGN ORDCMPLT ITEMORD ITEMNAME ITMPRICE F.STRAN
  GROUP BY CUSTNO AT LEVEL 1
  GROUP BY ORDERNO ORDRDATE ORPERSON ORDPONUM,
           ORDDUDAT ORDINVGN ORDCMPLT AT LEVEL 2
  ITEM CUSTNO,
       SPACES 0
  ITEM ORDPONUM ORDDUDAT ORDINVGN ORDCMPLT,
       ITEMORD ITEMNAME ITMPRICE,
       SPACES 1
  TITLE 'THE UPDATED CUSTOMER MASTER FILE RECORDS'
END REPORT
;
ENDPROCM: END PROC ;UPDATRPT
```

Figure 15-18 Type M Procedure UPDATRPT

The source listing of the UPDATRPT Procedure is in Type M Procedure UPDATRPT on page D-164.

In procedure UPDATRPT we need to complete the update and report the customer file updates. If the transaction record that just updated the customer file was the 5ADDITEM, we need to fill in the ITEMNAME and ITMPRICE in the new customer master record item segment. The item name and price are stored in the Item File which is indexed by the item number.

The 5ADDITEM transaction record just added a new item segment under an order segment in our matching customer record. Our Type M procedure can reference the new master work area item segment without looping. We will be referring to the one specific item occurrence just added.

To get the item name and price from the Item File, we need only reference a field in the Item File (Cord1). In the procedure, IF 1.ITEMNO = 1.ITEMNO command initiates the direct read of the Item File. The ITM5ORD1 field value from the

transaction record work area is used as the direct read key value. A true condition on the IF means that we found a matching item file record. We can then move the item name and price into the new master ITEMNAME and ITMPRICE fields.

If no matching item file record, we need to backout (remove) the item segment update. Since we cannot automatically reject the transaction at this point in the transaction cycle, we will procedurally delete the just created item segment in the new master work area. First we call the REJRPTS sub-procedure to record the error condition on our rejected transaction reports. Next, we clear the item segment key ITEMORD field and set the DELETE flag to 2. Finally, we GO TO the end of this Type M procedure, bypassing the reporting of the updated customer record.

For all valid updates, we produce a list of the customer record updates in a report. We will only report data from the segment occurrences matched by the current transaction record. This will be either the order segment of the item segment for the 3ADDORDR or the 5ADDITEM transaction records.

Immediately following all Type M procedures is one point where user-specified master file record or segment deletes will take place. When the DELETE flag is procedurally set to 1, the current master record is deleted. When the DELETE flag is procedurally set to 2, all marked segment occurrences are deleted. A segment occurrence is marked for deletion by placing blanks or zero into the segment key field or fields. In our update application, this is where item segments may get deleted.

**Note:** This method for procedurally deleting new master records or segments can be done in Type 2, 3, M, and N procedures. Deletion occurs after Type M, 3 or N time, whichever occurs first following the procedural setting of the DELETE flag.

Following all the Type M procedures, the Type 3 procedures are executed. Type 3 time is just like Type M time except that looping will occur on all referenced segment types. The segment path locking does not take place in Type 3 procedures.

In our update application, the Type 3 procedure ITEMLOOP is executed. See Figure 15-19.

```
ITEMLOOP: PROC TYPE3 BACKBRANCH 5
;
; Executed after a Transaction Record has updated the Customer Master
;  The Entire Master Record (all segment occurrances) is available
;
; We are looking for 5ADDITEM Transaction Records
;  For the ITEM Tran Rcds we need to manipulate Item/Qty Sets
;
IF X.TRANID = '5ADDITEM'
 CALL PROCEDURE ITEMCHKR             ; Go Check Item entries
 IF T.RETRYSW = 'YES'                ; Need to RETRY Tran
   TRANSFER TYPE_1                   ; Yes, Go back to TYPE 1
 END IF
END IF
END PROC ;ITEMLOOP
```

Figure 15-19 The Type 3 Procedure ITEMLOOP

The source listing of the ITEMLOOP Procedure is in Type 3 Procedure ITEMLOOP on page D-166.

In procedure ITEMLOOP, we need to manipulate the 5ADDITEM transaction records so we can process the multiple sets (up to 5 sets) of item number and quantity values.

The ITMCOUNT field is used to keep a count of which item number and quantity set has been processed. The 5ADDITEM transaction definition is defined so the item number and quantity data is processed from the set 1 positions within the transaction record. The first set of item number and quantity data is processed when the 5ADDITEM transaction record is read. For sets 2 through 5, we CALL a sub-procedure ITEMCHKR to check if there is data in the other sets to be processed. The data in sets 2 through 5 get moved to the set 1 positions in the ITEMCHKR sub-procedure. See Figure 15-20 for a look at sub-procedure ITEMCHKR.

```
ITEMCHKR: PROC SUBROUTINE
;
; For the 5ADDITEM Transaction Records
;  There are up to 5 ITEMORDER IDs and Quanity sets in the record
;  We must move the next set to the "first" set fields and RETRY
;
LET T.RETRYSW  = 'NO '
;
  IF T.ITMCOUNT = 1                   ; 1st Item/Qty Processed
    IF X.ITM5ORD2 NE ' '              ; So chk if 2nd Item/Qty Set
      LET X.ITM5ORD1 = X.ITM5ORD2     ; If yes, move data to theyt
      LET X.ITM5QTY1 = X.ITM5QTY2     ;  1st set positions
      LET T.RETRYSW  = 'YES'          ; Set Retry Switch
      RETURN                          ; Exit the Subroutine
    ELSE                              ;
      LET T.ITMCOUNT = T.ITMCOUNT + 1 ; If no, increment Item count
    END IF
  END IF
;
  IF T.ITMCOUNT = 2                   ; 2nd Item/Qty Processed
    IF X.ITM5ORD3 NE ' '              ; So chk if 3rd Item/Qty Set
      LET X.ITM5ORD1 = X.ITM5ORD3     ; If yes, move data to the the
      LET X.ITM5QTY1 = X.ITM5QTY3     ;  1st set positions
      LET T.RETRYSW  = 'YES'          ; Set Retry Switch
      RETURN                          ; Exit the Subroutine
    ELSE                              ;
      LET T.ITMCOUNT = T.ITMCOUNT + 1 ; If no, increment Item count
    END IF
  END IF
;
  IF T.ITMCOUNT = 3                   ; 3rd Item/Qty Processed
    IF X.ITM5ORD4 NE ' '              ; So chk if 4th Item/Qty Set
      LET X.ITM5ORD1 = X.ITM5ORD4     ; If yes, move data to the the
      LET X.ITM5QTY1 = X.ITM5QTY4     ;  1st set positions
      LET T.RETRYSW  = 'YES'          ; Set Retry Switch
      RETURN                          ; Exit the Subroutine
    ELSE                              ;
      LET T.ITMCOUNT = T.ITMCOUNT + 1 ; If no, increment Item count
    END IF
  END IF
;
  IF T.ITMCOUNT = 4                   ; 4th Item/Qty Processed
    IF X.ITM5ORD5 NE ' '              ; So chk if 5th Item/Qty Set
      LET X.ITM5ORD1 = X.ITM5ORD5     ; If yes, move data to the the
      LET X.ITM5QTY1 = X.ITM5QTY5     ;  1st set positions
      LET T.RETRYSW  = 'YES'          ; Set Retry Switch
      RETURN                          ; Exit the Subroutine
    ELSE                              ;
```

Figure 15-20 The Sub-Procedure ITEMCHKR (Page 1 of 2)

```
ITEMCHKR: PROC SUBROUTINE
      LET T.ITMCOUNT = T.ITMCOUNT + 1 ; If no, increment Item count
    END IF
  END IF
;
IF T.ITMCOUNT = 5           ; ALL 5 ITEM/QTY SETS PROCESSED
  LET T.ITMCOUNT = 0        ; YES, WE ARE DONE
  LET T.RETRYSW  = 'NO '    ;
END IF
END PROC ;ITEMCHKR
```

Figure 15-20 The Sub-Procedure ITEMCHKR (Page 2 of 2)

The source listing of the ITEMCHKR Procedure is in Sub-Procedure ITEMCHKR on page D-169.

In sub-procedure ITEMCHKR, we set the RETRYSW to 'NO'. Next we test the ITMCOUNT field value to see which one of the five (5) item number and quantity sets was just processed. Based on which set was just processed, we look at the next set to see if data (another item number value) is present.

When the next set of item number and quantity is empty, we add 1 to the ITMCOUNT and continue the checking with the next set.

When all five sets have been checked and processed, we set ITMCOUNT to zero, set the RETRYSW to 'NO', and end the sub-procedure, which returns to the calling procedure.

When more data is present, we move the item number and quantity values to the set 1 positions within the transaction record work area (ITM5ORD1 and ITM5QTY1), set the RETRYSW to 'YES' and RETURN to the calling procedure.

When the Type 3 procedure ITEMLOOP gets control back from the ITEMCHKR sub-procedure, the RETRYSW is tested to determine if there is more data to be processed. When the ITEMCHKR sub-procedure returns a 'YES' in the RETRYSW field, there are new data values in the set 1 positions, so we TRANSFER control back to TYPE_1 time to process (retry) the transaction record again.

When the RETRYSW contains a 'NO', there is no more item number and quantity data to be processed, so we are finished with the ITEMLOOP Type 3 procedure.

Following Type 3 procedures, the transaction processing cycle continues by reading the next transaction record.

When transactions are rejected, the transaction processing cycle jumps to Type 4 time. Transactions can be rejected automatically based on the transaction definition specifications. This can occur after Type 1 time (the transaction cannot be identified) or during the updating process following Type 2 time (alignment or validation issues). Transactions can be procedurally rejected by actions initiated during Type 1 and Type 2 procedures. In either case, control is given to Type 4 time.

In our update application, the Type 4 procedure TRANERR is executed whenever a transaction record is rejected. See Figure 15-21 for a look at the Type 4 procedure TRANERR.

```
TRANERR: PROC TYPE4 BACKBRANCH 5
;
; Executed when a Transaction is rejected
;  Automatically based on the Transaction Definition Criteria
;  or Procedurally by the Application Processing Criteria
;
IF F.XTRAN = 1
  CALL REPORT REJRPTS       ;Report Procedurally Rejected Transactions
ELSE
  LET T.ERRNOTE = 'AUTO REJECT, SEE DIAG MSG'
  CALL REPORT REJRPTS       ;Report Automatically Rejected Transactions
END IF
;
IF X.TRANID = '5ADDITEM'              ; Item ADD Trans,
 CALL PROCEDURE ITEMCHKR              ; Go Check Item entries
 IF T.RETRYSW = 'YES'                 ; Need to RETRY Tran
   TRANSFER TYPE_1                    ; Yes, Go back to TYPE 1
 END IF
END IF
END PROC ;TRANERR
```

Figure 15-21 The Type 4 Procedure TRANERR

The source listing of the TRANERR Procedure is in Type 4 Procedure TRANERR on page D-166.

In the TRANERR procedure, we want to report the rejected transactions. The ERRNOTE field is used to contain a brief comment about the rejected transaction. If the XTRAN flag contains a value of 1, the transaction was rejected procedurally and ERRNOTE was already filled in. Otherwise, the transaction was rejected automatically based on the transaction definition specifications, and a comment is placed in the ERRNOTE Field. In either case, we CALL a sub-procedure named REJRPTS to list the transaction record that was rejected. See Figure 15-22 for a look at sub-procedure REJRPTS.

```
REJRPTS: PROC TYPE SUB
;
; The REJECTED TRANSACTION REPORTs
;  One report for order   transaction records

;  One report for item    transaction records

;  One report for unknown transaction records

;
IF X.TRANID = '3ADDORDR'
;
  REPORT X.TRANID X.CUSTNO X.ORDERNO X.ORDRDATE X.ORDRPERS X.ORDRPONO,
         X.ORDRDUED X.ORDRINVG X.ORDRCOMP F.XTRAN F.M4TRAN,
         T.ERRNOTE
    ITEM X.TRANID,
         SPACES 0
    ITEM X.CUSTNO X.ORDERNO X.ORDRDATE X.ORDRPERS X.ORDRPONO,
         X.ORDRDUED X.ORDRINVG X.ORDRCOMP F.XTRAN F.M4TRAN,
         SPACES 1
    ITEM F.M4TRAN,
         ENDLINE
    TITLE 'REJECTED TRANSACTION REPORT - ADD ORDER TRANSACTION RECORDS'
  END REPORT
END IF
;
IF X.TRANID = '5ADDITEM'
;
```

Figure 15-22 The Sub-Procedure REJRPTS (Page 1 of 2)

```
REJRPTS: PROC TYPE SUB
  REPORT X.TRANID X.CUSTNO X.ORDERNO X.ITM5ORD1 X.ITM5QTY1,
         X.ITM5ORD2 X.ITM5QTY2,
         X.ITM5ORD3 X.ITM5QTY3,
         X.ITM5ORD4 X.ITM5QTY4,
         X.ITM5ORD5 X.ITM5QTY5,
         F.XTRAN F.M4TRAN T.ERRNOTE
    ITEM X.TRANID,
         SPACES 0
    ITEM X.CUSTNO X.ORDERNO X.ITM5ORD1 X.ITM5QTY1 X.ITM5ORD2 X.ITM5QTY2,
         X.ITM5ORD3 X.ITM5QTY3 X.ITM5ORD4 X.ITM5QTY4,
         X.ITM5ORD5 X.ITM5QTY5 F.XTRAN F.M4TRAN,
         SPACES 1
    ITEM F.M4TRAN,
         ENDLINE
    TITLE 'REJECTED TRANSACTION REPORT - ADD ITEMS TRANSACTION RECORDS'
  END REPORT
END IF
;
IF X.TRANID NE '3ADDORDR' '5ADDITEM'
;
  REPORT F.M4TRAN X.TRANID X.ALL80
    TITLE 'REJECTED TRANSACTION REPORT - UNKNOWN TRANSACTION RECORDS'
  END REPORT
END IF
END PROC ;REJRPTS
```

Figure 15-22 The Sub-Procedure REJRPTS (Page 2 of 2)

The source listing of the REJRPTS Procedure is in Sub-Procedure REJRPTS on page D-167.

In the sub-procedure REJRPTS we produce three rejected transaction reports based the transaction identifier. The first report shows the rejected 3ADDORDR transaction records. The second report shows the rejected 5ADDITEM transaction records. The third report shows the rejected transaction records with an unknown transaction identifier.

When we return to the TRANERR procedure after the reject information data is reported, we need to check on the 5ADDITEM transaction records.

Remember, we need to manipulate the 5ADDITEM transaction records so we can process the multiple sets (up to 5 sets) of item number and quantity values.

The ITMCOUNT field is used to keep a count of which item number and quantity set has been processed. The 5ADDITEM transaction definition is defined so the item number and quantity data is processed from the set 1 positions within the transaction record. The first set of item number and quantity data is processed when the 5ADDITEM transaction record is read. For sets 2 through 5, we CALL a sub-procedure ITEMCHKR to check if there is data is the other sets to be processed. The data in sets 2 through 5 get moved to the set 1 positions in the ITEMCHKR sub-procedure. See Figure 15-20 for a look at sub-procedure ITEMCHKR.

When the Type 4 procedure TRANERR gets control back from the ITEMCHKR sub-procedure, the RETRYSW is tested to determine if there is more data to be processed. When the ITEMCHKR sub-procedure returns a ‘YES’ in the RETRYSW field, there are new data values in the set 1 positions, so we TRANSFER control back to TYPE_1 time to process (retry) the transaction record again.

When the RETRYSW contains a 'NO', there is no more item number and quantity data to be processed, so we are finished with the TRANERR procedure.

Following Type 4 procedures, the transaction processing cycle continues by reading the next transaction record.

When all the transaction records for the current customer master file record have been processed, or if no transactions aligned with the customer record, we end the transaction processing cycle for the current customer master record. The new master record is now available for standard (Normal, Type N) processing outside of the transaction processing cycle.

In our updating application, the Type N (Normal) procedure is executed. See Figure 15-23.

```
MAIN: PROC TYPE NORMAL
;
; Executed for each Customer Master Record after ALL Updates (if any)
; have been completed.
;
REPORT CUSTNO ORDERNO ORDRDATE ORPERSON ORDPONUM ORDDUDAT,
       ORDINVGN ORDCMPLT ITEMORD ITEMNAME ITMPRICE F.TRAN
  GROUP BY CUSTNO F.TRAN AT LEVEL 1
  GROUP BY ORDERNO ORDRDATE ORPERSON ORDPONUM,
           ORDDUDAT ORDINVGN ORDCMPLT AT LEVEL 2
  ITEM CUSTNO,
       SPACES 0
  ITEM ORDPONUM ORDDUDAT ORDINVGN ORDCMPLT,
       ITEMORD ITEMNAME ITMPRICE,
       SPACES 1
  TITLE 'THE NEW CUSTOMER MASTER FILE RECORDS'
END REPORT
END PROC
```

Figure 15-23 The Type N (Normal) Procedure MAIN

The source listing of the MAIN Procedure is in Type N (Normal) Procedure MAIN on page D-171.

The MAIN procedure produces a report of all the customer master file records.

This completes the description of the updating application run. The complete listing of the updating application coding starts in Run Control Commands on page D-161.

Now we can look at the output reports produced by the update application. The complete listing of the updating application outputs starts in Listing of Transaction Records Report on page D-172.

Before you view the reports produced by the update application run, you can look at the report of the customer master file records that were input to the update run. This report was produced by the application example referenced in Chapter 10, *Transactions* that created the customer master file. The report is in Listing of CUSTMAST Contents on page D-160.

# The Update Application Reports and Outputs

All the Reports and Outputs displayed in the section are also in Appendix D following the update application source listing. These reports and outputs start at Transaction Diagnostic Messages on page D-171.

Here is a report showing a simple listing of the transaction records input to the run. This report is produced in the Type 1 procedure TRANPREP. See Figure 15-24 for the report listing.

```
                                    LISTING OF TRANSACTION RECORDS 80/80 WITH RECORD COUNT                          PAGE 1
-------------------------------------------------------------------------------------------------------------------------------
      M4TRAN                                        TRAN RECORD
                                                  ALL 80 POSITIONS
-------------------------------------------------------------------------------------------------------------------------------
              1  3ADDORDR0000199000062001JOE12345030302NN
              2  5ADDITEM0000199000G46     00001M33     00003M56     00002D77     00001F74     00001
              3  3ADDORDR0000299001062001BOB6789A120401NN
              4  5ADDITEM0000299001H98     00003ERR     00099
              5  3ADDORDR0000299002062101APCB531A020502NN
              6  5ADDITEM0000299002G46     00002
              7  3ADDORDR0001A99000062001ERR12345030302NN
              8  3ADDORDR0011599014082701RAN12543043002NN
              9  5ADDITEM0011599014L99     00003013     00002M56     00000
             10  3ADDORDR0011599914082701999125430430 02NN
             11  3ADDORDR0034399018062001KENB1632070601NN
             12  5ADDITEM0034399018K69     00001
             13  3ADDORDR0063899006062794NOM99953020102NN
             14  3ADDORDR0073199012052901RANB3219030302NN
             15  4ADDITEM0073199012019     00003872     00002
             16  5ADDITEM0073199012019     00003872     00002
             17  3ADDORDR0073199019071701RANC1399101301NN
             18  5ADDITEM0073199019027     00001072     00004ERR     00000
             19  3ADDORDR0073899006062794TOD99953020102NN
             20  5ADDITEM0077799006ERR     00009
             21  3ADDORDR0081899007052101JJKABC81091501NN
             22  5ADDITEM0081899007H61     00008041     00007
             23  3ADDORDR0081899020071501MBPBING1101901NN
             24  3ADDERRR0081899020071501MBPBING1101901NN
             25  5ADDITEM0081899020D77     00006E56     00005E66     00004
             26  3ADDORDR0081899920171501MBPBING1101901NN
             27  3ADDORDR0081899921071501MBPBING1131901NN
             28  3ADDORDR0084799008062101KREX0C81081501NN
             29  5ADDITEM0084799008F74     00003
             30  3ADDORDR0084799009062201JONX0C82081901NN
             31  5ADDITEM0084799009F91     00001M57     00002
             32  3ADDORDR0091899010071301BOBBC123111901NN
             33  5ADDITEM0091899010F87     00001L99     00003M50     00002
             34  3ADDORDR0100099022080501IKE99B51033102NN
             35  5ADDITEM0100099022072     00002
```

Figure 15-24 Listing of Transaction Records Input to the Update Run

The report shows the input record number and the entire 80-byte record data. You reference to this when looking at the Diagnostic Messages and Rejected Transaction Reports produced by the update run.

Here is the report of the updated customer records. The report only shows the customer record updates. This report is produced in the Type M procedure UPDATRPT. See Figure 15-25 for the report listing.

THE UPDATED CUSTOMER MASTER FILE RECORDS PAGE 1

| CUSTOMER NUMBER | ORDER NUMBER | ORDER DATE | SALES PERSON | PURCH. ORDER NUMBER | DUE DATE | INVOICE GENERATED? | ORDER COMPLETED? | STOCK NO. ORDERED | DESCRIPTION | PRICE | STRAN |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 00001 | 99000 | 062001 | JOE | 12345 | 030302 | N | N | - | - | $- | MI |
| | | | | | | | | G46 | V:TRAN UTILITY GDE | 15.00 | MMI |
| | | | | | | | | M33 | V:TRAN 7.5 CHG PAGES | 5.00 | MMI |
| | | | | | | | | M56 | 2" BINDER NOTEBOOK | 7.50 | MMI |
| | | | | | | | | D77 | V:TRAN CLASS GDE-A | 97.50 | MMI |
| | | | | | | | | F74 | V:TRAN CLASS GDE-C | 103.50 | MMI |
| 00002 | 99001 | 062001 | BOB | 6789A | 120401 | N | N | - | - | $- | MI |
| | | | | | | | | H98 | OB6789A120481NN8 | 15.00 | MMI |
| | 99002 | 062101 | APC | B531A | 020502 | N | N | - | - | $- | MI |
| | | | | | | | | G46 | V:TRAN UTILITY GDE | 15.00 | MMI |
| 00115 | 99014 | 082701 | RAN | 12543 | 043002 | N | N | - | - | $- | MI |
| | | | | | | | | L99 | V:BLDR ENVIRON GDE-O | 17.50 | MMI |
| | | | | | | | | 013 | V:BLDR REF SUMMARY | 51.00 | MMI |
| 00343 | 99018 | 062001 | KEN | B1632 | 070601 | N | N | - | - | $- | MI |
| | | | | | | | | K69 | EEO REPORT GUIDE | 67.00 | MMI |
| 00731 | 99012 | 052901 | RAN | B3219 | 030302 | N | N | - | - | $- | MI |
| | | | | | | | | 019 | V:BLDR ENV GUIDE-VSE | 51.00 | MMI |
| | | | | | | | | 872 | G/L REFERENCE GUIDE | 75.00 | MMI |
| | 99019 | 071701 | RAN | C1399 | 101301 | N | N | - | - | $- | MI |
| | | | | | | | | 027 | V:BLDR CLASS GUIDE-A | 55.00 | MMI |
| | | | | | | | | 072 | V:BLDR CLASS GUIDE-B | 55.00 | MMI |
| 00738 | 99006 | 062794 | TOD | 99953 | 020102 | N | N | - | - | $- | MI |
| 00818 | 99007 | 052101 | JJK | ABC81 | 091501 | N | N | - | - | $- | MI |
| | | | | | | | | H61 | V:BLDR INST GDE-VSE | 15.00 | MMI |
| | | | | | | | | 041 | V:BLDR PROCEDURE GDE | 55.00 | MMI |
| | 99020 | 071501 | MBP | BING1 | 101901 | N | N | - | - | $- | MI |
| | | | | | | | | D77 | V:TRAN CLASS GDE-A | 97.50 | MMI |
| | | | | | | | | E56 | V:BLDR USER I/O GDE | 27.50 | MMI |
| | | | | | | | | E66 | V:BLDR GDBI GUIDE | 37.50 | MMI |
| 00847 | 99008 | 062101 | KRE | X0C81 | 081501 | N | N | - | - | $- | MI |
| | | | | | | | | F74 | V:TRAN CLASS GDE-C | 103.50 | MMI |
| | 99009 | 062201 | JON | X0C82 | 081901 | N | N | - | - | $- | MI |
| | | | | | | | | F91 | V:TRAN ADM GUIDE | 30.00 | MMI |
| | | | | | | | | M57 | 3" BINDER NOTEBOOK | 9.50 | MMI |
| 00918 | 99010 | 071301 | BOB | BC123 | 111901 | N | N | - | - | $- | MI |
| | | | | | | | | F87 | V:TRAN CLASS GDE-B | 97.50 | MMI |
| | | | | | | | | L99 | V:BLDR ENVIRON GDE-O | 17.50 | MMI |
| | | | | | | | | M50 | V:TRAN SECURITY GDE | 27.50 | MMI |
| 01000 | 99022 | 080501 | IKE | 99B51 | 033102 | N | N | - | - | $- | MI |
| | | | | | | | | 072 | V:BLDR CLASS GUIDE-B | 55.00 | MMI |

Figure 15-25 The Updated Customer Master File Records Report

**Note:** The report in Figure 15-25 is produced without looping on the segments referenced by the field names being reported. Since the report is produced at Type M time, only the segment occurrences matched (or inserted) by the current transaction record are available for reporting. Each line in the report represents data in the new work area for each transaction record pass through the Type M procedure UPDATRPT.

The STRAN Flag settings indicate matches (M) and inserts (I) at segment levels. The setting of MI is indicating a match at level 1 on customer number, and an insert at level 2 for the added order segment occurrence. The setting of MMI is indicating a match at level 1 on customer number, a match at level 2 on the order number, and an insert at level 3 for the added item segment occurrence.

Here is the report of the orders with order dates earlier that the master file order dates. This report is produced in the Type 2 procedure ORDERCHK. See Figure 15-26 for the report listing.

```
                         LIST OF ORDERS WITH ORDER DATES EARILER THAN MASTER FILE ORDER DATES                PAGE 1
-------------------------------------------------------------------------------------------------------------------------
  TRANS    CUST   ORDER     TRAN       MASTER        M4TRAN
   ID      NO.     NO.      DATE        DATE
-------------------------------------------------------------------------------------------------------------------------
 3ADDORDR  00001  99000  06/20/2001  08/01/2001              1
 3ADDORDR  00343  99018  06/20/2001  07/11/2001             11
 3ADDORDR  00343  99018  06/20/2001  08/01/2001             11
 3ADDORDR  00738  99006  06/27/1994  01/14/2001             19
```

Figure 15-26 The Order Dates that Pre-date Master File Order Dates Report

The report shows the order transaction record information and master file order dates. Whenever an order transaction contains an order date that is earlier that any order date on the master file for the customer, a line is produced on this report. You can see that the transaction record for order number 99018 and customer 00343 had an order date of 06/20/2001 which was earlier than two other order dates currently in the customer master file record for customer 00343.

Now we will look at the information and reports produced by the update run for the rejected transaction records. There are three separate rejected transaction reports. See Figure 15-27 through Figure 15-29. The first report shows the rejected add order (3ADDORDR) transactions, the second report shows the rejected add item (5ADDITEM) transactions, and the third report shows the rejected transactions that are unknown to the update application.

```
                         REJECTED TRANSACTION REPORT - ADD ORDER TRANSACTION RECORDS                    PAGE 1
-------------------------------------------------------------------------------------------------------------------------
 TRANS   CUST  ORDER ORDER  ORDER  ORDER  DUE   INV INV XTRAN     M4TRAN
  ID     NO.    NO.  DATE   PERSON  PO    DATE  GEN CMP
-------------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------------------------
          ERROR
         COMMENT
-------------------------------------------------------------------------------------------------------------------------
3ADDORDR 0001A 99000 062001  ERR   12345 030302  N   N     7              7
  AUTO REJECT, SEE DIAG MSG
3ADDORDR 00115 99914 082701  999   12543 043002  N   N     7             10
  AUTO REJECT, SEE DIAG MSG
3ADDORDR 00638 99006 062794  NOM   99953 020102  N   N     8             13
  AUTO REJECT, SEE DIAG MSG
3ADDORDR 00818 99920 171501  MBP   BING1 101901  N   N     6             26
  AUTO REJECT, SEE DIAG MSG
3ADDORDR 00818 99921 071501  MBP   BING1 131901  N   N     6             27
  AUTO REJECT, SEE DIAG MSG
```

Figure 15-27 The Transaction Report for Add Order Records

```
                         REJECTED TRANSACTION REPORT - ADD ITEMS TRANSACTION RECORDS                    PAGE 1
-------------------------------------------------------------------------------------------------------------------------
 TRANS   CUST  ORDER  ITEM    ITEM    ITEM    ITEM    ITEM    ITEM    ITEM    ITEM    ITEM    ITEM   XTRAN     M4TRAN
  ID     NO.    NO.  ORDR 1  QTY 1  ORDR 2  QTY 2  ORDR 3  QTY 3  ORDR 4  QTY 4  ORDR 5  QTY 5
-------------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------------------------
          ERROR
         COMMENT
```

Figure 15-28 The Transaction Report for Add Item Records (Page 1 of 2)

```
-----------------------------------------------------------------------------------------------------------------------------------
5ADDITEM 00002 99001 ERR          99 ERR          99                                                                    4
  ITEM ORDERED NOT ON FILE
5ADDITEM 00115 99014 M56             013           2 M56                                                  1              9
  QTY INVALID FROM SET 3
5ADDITEM 00731 99019 ERR             072           4 ERR                                                  1             18
  QTY INVALID FROM SET 3
5ADDITEM 00777 99006 ERR           9                                                                     8             20
  AUTO REJECT, SEE DIAG MSG
```

Figure 15-28 The Transaction Report for Add Item Records (Page 2 of 2)

```
                                  REJECTED TRANSACTION REPORT - UNKNOWN TRANSACTION RECORDS                          PAGE 1
-----------------------------------------------------------------------------------------------------------------------------------
      M4TRAN       TRANS                                        TRAN RECORD
                    ID                                        ALL 80 POSITIONS
-----------------------------------------------------------------------------------------------------------------------------------
              15  4ADDITEM  4ADDITEM0073199012019    00003872    00002
              24  3ADDERRR  3ADDERRR0081899020071501MBPBING1101901NN
```

Figure 15-29 The Transaction Report for Unknown Records

On the rejected transaction reports, there is error comment text to help determine the reason for the reject. The report of Unknown Transactions does not need the error comment.

The AUTO REJECT, SEE DIAG MSG indicates that the transaction was rejected automatically. There is more information on these in the Transaction Diagnostic Messages output from the update run, shown in Figure 15-30.

For procedurally-rejected transactions, there is text that indicates why the application rejected the transaction. Remember that add item transaction records can contain up to five sets of item number and quantity values. The error comment text will indicate which set was in error. Only that set was rejected, not the entire record.

The XTRAN Flag value provides the reason a transaction record was rejected. A value of:

- 1 indicates procedural rejection.
- 6 indicates a date validation error.
- 7 indicates a pattern validation error.
- 8 indicates that the transaction did not match the master.

Refer to the *VISION:Builder ASL Reference Guide* or the *VISION:Builder Reference Summary* for a complete list of the XTRAN Flag settings.

On the rejected add item report in Figure 15-28, the first reject record shows a blank (zero setting) under the XTRAN column. Since this record was rejected in our Type M procedure UPDATRPT, the XTRAN Flag is not active at that point in the transaction processing cycle. Remember, at Type M time, the transaction has already updated (inserted) the item segment. The UPDATRPT procedure attempted to get the item name and price, but none was found. The REJRPT sub-procedure was called to list the error condition and the item segment was dynamically deleted.

The M4TRAN Flag values are listed. This shows the current value of the number of records input (read) from the transaction file. You can use this to refer back to the listing of the transaction reports as shown in Figure 15-24. Also, this value can be used to reference the Transaction Diagnostic Messages output from the update run. See Figure 15-30.

```
** MK4TG05  TYPE 1  TRANSACTION IDENTIFIER 3ADDORDR - FIELD VIOLATES INPUT EDIT PATTERN SPECIFICATIONS. FIELD LOC     9       *****
                   M4TRAN   RECORD        7 - KEY 1 - 0001A
** MK4TG05  TYPE 1  TRANSACTION IDENTIFIER 3ADDORDR - FIELD VIOLATES INPUT EDIT PATTERN SPECIFICATIONS. FIELD LOC    25       *****
                   M4TRAN   RECORD       10 - KEY 1 - 00115
** MK4RL01  TYPE 1  TRANSACTION IDENTIFIER 3ADDORDR - RECORD NOT ON MASTER FILE AND CREATE NOT SPECIFIED.                    *****
                   M4TRAN   RECORD       13 - KEY 1 - 00638
                   M4OLD    RECORD        8 - KEY 1 - 00731
                   M4NEW    RECORD        7 - KEY 1 - 00551
** MK4RL01  TYPE 1  TRANSACTION IDENTIFIER 5ADDITEM - RECORD NOT ON MASTER FILE AND CREATE NOT SPECIFIED.                    *****
                   M4TRAN   RECORD       20 - KEY 1 - 00777
                   M4OLD    RECORD       10 - KEY 1 - 00818
                   M4NEW    RECORD        9 - KEY 1 - 00738
** MK4TG06  TYPE 1  TRANSACTION IDENTIFIER 3ADDORDR - DATE FIELD IS INVALID. FIELD LOC IS    19                             *****
                   M4TRAN   RECORD       26 - KEY 1 - 00818
** MK4TG06  TYPE 1  TRANSACTION IDENTIFIER 3ADDORDR - DATE FIELD IS INVALID. FIELD LOC IS    33                             *****
                   M4TRAN   RECORD       27 - KEY 1 - 00818
```

Figure 15-30 The Transaction Diagnostic Output Messages from the Update Run

This information is listed by VISION:Builder whenever transactions are automatically rejected based on the transaction definition specifications. The M4TRAN RECORD number can be used to reference the rejected transaction reports.

After all the updating has been performed, we have the new updated customer master file. The last report produced by our update run is the list of all the customer records on the new output master file. See Figure 15-31.

```
                                        THE NEW CUSTOMER MASTER FILE RECORDS                                        PAGE 1
-------------------------------------------------------------------------------------------------------------------------
CUSTOMER  ORDER   ORDER   SALES  PURCH. ORDER  DUE     INVOICE     ORDER     STOCK NO.     DESCRIPTION          PRICE    TRAN
 NUMBER   NUMBER  DATE    PERSON   NUMBER     DATE  GENERATED? COMPLETED?  ORDERED
-------------------------------------------------------------------------------------------------------------------------
00001     10001   010401  PEG      A0001    100101     N          N        G38      1" BINDER NOTEBOOK      $5.00    1
                                                                           G40      V:BLDR REFERENCE GDE     35.00
          10002   080101  BRR      A0002    110101     N          N        M14      V:BLDR ASL REF GDE      $31.00
                                                                           M23      V:TRAN SCREEN GDE        25.00
          99000   062001  JOE      12345    030302     N          N        D77      V:TRAN CLASS GDE-A      $97.50
                                                                           F74      V:TRAN CLASS GDE-C      103.50
                                                                           G46      V:TRAN UTILITY GDE       15.00
                                                                           M33      V:TRAN 7.5 CHG PAGES      5.00
                                                                           M56      2" BINDER NOTEBOOK        7.50
00002     99001   062001  BOB      6789A    120401     N          N        H98      OB6789A120481NN8        $15.00    1
          99002   062101  APC      B531A    020502     N          N        G46      V:TRAN UTILITY GDE      $15.00
00115     11501   071201  PEG      A1151    073001     Y          Y        010      ACCOUNTS PAYABLE UM     $45.00    5
          11502   072201  PEG      A1152    083101     Y          N        072      V:BLDR CLASS GUIDE-B    $55.00
                                                                           106      V:BLDR TABLE TIPS         5.00
                                                                           110      V:BLDR REPORT TIPS        5.00
          99014   082701  RAN      12543    043002     N          N        L99      V:BLDR ENVIRON GDE-O    $17.50
                                                                           013      V:BLDR REF SUMMARY        51.00
00131     31135   050501  JAM      K123J    070101     Y          Y        D77      V:TRAN CLASS GDE-A      $97.50
          31735   070701  JAM      K132J    080101     Y          Y        027      V:BLDR CLASS GUIDE-A    $55.00
00294     29401   042901  JEE      B2941    060101     Y          Y        978      PURCHASE ORDER GDE      $35.00
00343     34301   071101  ANS      A3431    080301     Y          Y        865      V:INFM REFERENCE GDE    $39.00    1
                                                                           866      V:INFM INSTALL GDE       19.00
                                                                           873      V:INFM OPS GUIDE         19.00
          34302   080101  PEG      A3423    090501     N          N        852      PURCHASE ORDER ENVR     $24.00
                                                                           872      G/L REFERENCE GUIDE      75.00
          99018   062001  KEN      B1632    070601     N          N        K69      EEO REPORT GUIDE        $67.00
00551     51135   060601  JAM      K123J    070101     Y          Y        D77      V:TRAN CLASS GDE-A      $97.50
          51755   070901  JAM      K132J    080901     Y          Y        027      V:BLDR CLASS GUIDE-A    $55.00
00731     99012   052901  RAN      B3219    030302     N          N        019      V:BLDR ENV GUIDE-VSE    $51.00    1
                                                                           872      G/L REFERENCE GUIDE      75.00
          99019   071701  RAN      C1399    101301     N          N        027      V:BLDR CLASS GUIDE-A    $55.00
                                                                           072      V:BLDR CLASS GUIDE-B     55.00
```

Figure 15-31 The Report of the New Customer Master File Records (Page 1 of 2)

```
 00738    73891   011401   JEF      B7381     012901     Y          Y       L06      V:INFM CLASS PACK      $299.99    1
                                                                            L06F     V:INFM CLASS SLIDES      99.00
                                                                            L96      V:BLDR APPL GUIDE        50.00
          99006   062794   TOD      99953     020102     N          N            -                    -        $-
 00818    99007   052101   JJK      ABC81     091501     N          N       H61      V:BLDR INST GDE-VSE     $15.00    5
                                                                            041      V:BLDR PROCEDURE GDE     55.00
          99020   071501   MBP      BING1     101901     N          N       D77      V:TRAN CLASS GDE-A      $97.50
                                                                            E56      V:BLDR USER I/O GDE      27.50
                                                                            E66      V:BLDR GDBI GUIDE        37.50
 00847    99008   062101   KRE      X0C81     081501     N          N       F74      V:TRAN CLASS GDE-C     $103.50    1
          99009   062201   JON      X0C82     081901     N          N       F91      V:TRAN ADM GUIDE        $30.00
                                                                            M57      3" BINDER NOTEBOOK        9.50
                                                THE NEW CUSTOMER MASTER FILE RECORDS                                    PAGE 2
-------------------------------------------------------------------------------------------------------------------------------
CUSTOMER  ORDER   ORDER   SALES  PURCH. ORDER  DUE    INVOICE     ORDER    STOCK NO.     DESCRIPTION        PRICE    TRAN
 NUMBER   NUMBER  DATE    PERSON    NUMBER     DATE  GENERATED? COMPLETED?  ORDERED
-------------------------------------------------------------------------------------------------------------------------------
 00867    86731   061501   WAV      W1531     100101     Y          Y       013      V:BLDR REF SUMMARY      $51.00
          86751   071501   SUR      S1552     110901     Y          Y       041      V:BLDR PROCEDURE GDE    $55.00
 00918    91801   051201   GEE      A9181     071501     Y          Y       L97      V:INFM QUERY GUIDE      $45.00    1
                                                                            M01      V:TRAN ENV GDE CICS      47.50
          99010   071301   BOB      BC123     111901     N          N       F87      V:TRAN CLASS GDE-B      $97.50
                                                                            L99      V:BLDR ENVIRON GDE-O     17.50
                                                                            M50      V:TRAN SECURITY GDE      27.50
 00951    91395   090601   JOH      9123A     010902     N          N       013      V:BLDR REF SUMMARY      $51.00
          91795   090901   UNC      9132B     010902     N          N       041      V:BLDR PROCEDURE GDE    $55.00
 01000    99022   080501   IKE      99B51     033102     N          N       072      V:BLDR CLASS GUIDE-B    $55.00    1
```

Figure 15-31 The Report of the New Customer Master File Records (Page 2 of 2)

This report shows all the customer records on file. The TRAN Flag indicates which records were updated in the run. A blank (zero) under the TRAN column indicates that there were no transaction records for the master. A value of 1 in the TRAN Flag indicates that the master file record was updated or matched by a transaction record. A value of 5 in the TRAN Flag indicates a setting of 1 plus 4. Along with the master file record being updated or matched by a transaction record (1), at least one transaction record was rejected for the master record (4).

Finally, each VISION:Builder processing run automatically provides the input/output record statistics. See Figure 15-32 for the statistics from our update run.

```
** MK4W204  TYPE 0  NUMBER OF MESSAGES PRINTED IS    6.                                                                  *****
                    M4OLD    -       15 RECORDS INPUT
                    M4NEW    -       15 RECORDS OUTPUT
                    M4TRAN   -       35 RECORDS INPUT
                    M4CORD1  -       26 RECORDS INPUT
                    M4REPO   -      229 RECORDS OUTPUT
                    M4INPUT  -      277 RECORDS INPUT
                    M4LIST   -      869 RECORDS OUTPUT
                      15 TRACKS ASSIGNED TO M4LIB    --   13 TRACKS NOT FULL. LIBRARY DIRECTORY BLOCKING FACTOR IS  800.
** MK4W209  TYPE 0  1002840 BYTES OF MAIN STORAGE UNUSED DURING DECODING PHASE                                          *****
                     997038 BYTES OF MAIN STORAGE UNUSED DURING COMPILATION PHASE
                     564944 BYTES OF MAIN STORAGE UNUSED DURING PROCESSING PHASE
```

Figure 15-32 Statistics From the Update Run

# Transaction and Master Files for Applications Other than Updating

There are application problems that are best solved by assigning your application's primary data file to the VISION:Builder transaction file. This allows you to let automatic transaction processing build a (temporary) data structure in the new master work area. When you use the transaction and master files in this way (the transaction master technique), you rarely output the new master file. The

technique uses the capabilities of transaction definitions to build the new master work area. See Chapter 6, *Input File Processing* and Chapter 10, *Transactions* for more details.

### Pre-Summarization

To illustrate processing a sequential file containing a large number of records, think of a file of time card records for a large manufacturing organization. In an application, we need to locate all employees who charged more than 200 hours to product A, more than 150 hours to product D, or more than 300 hours to product C. This application could be constructed entirely with type N procedures, including the statements to summarize time for the three products for each employee into temporary fields, as well as the statements to check for a change in employee number.

However, the better alternative is the transaction master technique. The transaction master technique allows you to process and report from detail records and summarize data from them in a single pass of your application's data file, saving CPU time. You need to write one or two file definitions, a transaction definition, and type 3 and N procedures.

- The master file work area's file definition can be as simple or as complex as your summarization requirements dictate. If you require only one level of summary, define a work area with only a root segment; if you require several levels of summary action, define an appropriate hierarchy. Record and segment keys are those fields that define each level of summarization. In the case of the time card task, a file definition is used for a single level file, where the key is employee number. The definition also needs fields in which to accumulate hours charged, one each for products A, D, and C.
- The transaction definition specifies the rules for building summary new master work areas from the data file. For the time card task, define three simple transactions. Product code serves as the identifier; each transaction definition adds the time card hours into the appropriate product's field in the master file work area. The transactions matches on employee number with default create record equals Y. Every time a new employee number is read on the transaction file, VISION:Builder stops adding time card hours for the previous employee, executes type N procedures, and starts the transaction cycle again for the next employee.
- The transaction file definition is a straightforward definition of the data file. You can use this definition in type 3 procedures for processing and reporting from the individual detail records in your data file.
- Type N procedures process and report from the summarized data in the new master work area. For the time card task, a type N procedure checks to see if the employee's charges to products A, D, or C were great enough to include that employee on the report.

Finally, sort the time card file by the key field (employee number) of the root segment in the new master work area.

### Eliminating Duplicate Keys

If you have a coordination problem where all files involved have duplicate records per key value, you will be able to solve the problem using the transaction master technique with one of the files. In this case, define your master record work area's hierarchy so that the root segment contains only the record key. The single level two segment type contains an entire record. All records with the same key are built into one hierarchy. In this situation, type 1 or 3 procedures are not needed because automatic transaction processing does the entire job of building the temporary hierarchy.

### Header/Trailer Record Processing

If you need to process a sequential file containing header and trailer records, use the transaction master technique. The use of the technique is similar to that just described for pre-summarization.

- The file definition for the new master work area is hierarchically structured. Define the root segment to hold the contents of a header record. Define a lower level segment for each type of trailer record. Assign parentage of these lower level segments according to the relationships between the header and trailer records and between the various types of trailer records.
- Prepare transaction definitions, one for each type of header and trailer record. If values for use as identifiers are not apparent, remember that a transaction identifier can be a logical concatenation of two values or the absence of a value. If your trailer records do not contain the header record's primary key value, define an appropriate length as a part of the trailer's identifier to hold the header's key field, even if it overlays another field. How to manage the situation is described below.
  - Write a file definition for the application data file of header and trailer records if you need to specify any type 1, 2, 3, or 4 procedure processing. You must do this if you have the problem of incomplete keys in trailer records. In this file definition, you will be defining the same area several times, once for the header and once for each type of trailer record.
  - If your trailer records do not contain the primary key from the header, prepare a type 1 procedure. During a transaction cycle when a header record is in the transaction file work area, store the primary key value in a temporary field. During any transaction cycle when a trailer record is in the transaction file work area, move the primary key from the temporary field to the specially defined field in the transaction work area. If this overlays a data field in the trailer record, first save the contents of that field in a second temporary field and move the contents of this temporary field to the appropriate field in the new master work area in a type M procedure.

During the type N procedure portion of the master record time frame, you have one set of header and trailer records in the form of a hierarchy in the new master work area.

The header/trailer information is now organized and related. This could become your master file, or you could output the information to a subfile as reformatted and organized transaction records.

There is an example of a simple Header and Trailer Record Transaction Processing Application in Appendix D, *Complete Sample Applications*. There are three definitions referenced in the Header Trailer Application. The definition names are: HDTRMSTR (master file being created), HDTRATRN (transaction file definition), and HDTRGRP (the transaction group definition). The definitions are in HDTRMSTR File Definition on page D-150, HDTRATRN File Definition on page D-152, and HDTRGRP Transaction Group Definition on page D-153 respectively.

The Header Trailer Application and Report Output is in Source Code and Report Output on page D-176.

## IMS Updating Considerations

Before updating IMS databases, you should familiarize yourself with the relationship of VISION:Builder to the database definition (DBD). This is described in Chapter 3, *Defining Data*. If the IMS database is very large, you should pay careful attention to the discussion of MOSAIC processing in Chapter 16, *Database Access Methodology*.

While VISION:Builder can update logical records in work areas with a great deal of flexibility, IMS restricts updating when the database is physically modified. The restrictions are explicitly coded as insert, delete, and replace rules in the DBD. The action code on your transaction definitions must be consistent with these explicit codes. For example, VISION:Builder can insert segment occurrences in logical records that are subsequently rejected by IMS because the explicit insert code in the DBD requires an existing related segment occurrence.

## Relational Updating Considerations

Chapter 10, *Transactions* introduced the transaction processing cycle and the techniques provided by VISION:Builder to maintain application files. Transaction processing, using transaction definitions cataloged in the common library against a file definition, automatically and reliably updates the file. To update, indicate the master file requiring maintenance, supply the file definition of the transaction records, and specify the transaction definitions relating them.

This process is the same whether you are updating a sequential file or a relational database. All the concepts of transaction processing apply without change when updating a master file that is a relational database.

This section focuses on the special considerations of updating a relational database with VISION:Builder, such as the potential for updating a set of rows, referential integrity constraints imposed by the relational database manager, and the order in which VISION:Builder performs inserts, deletes, and updates.

The updating of a relational database is supported for update-in-place master files using either standard (non-MOSAIC) processing or memory optimized processing (MOSAIC).

- With non-MOSAIC processing, you can only use a single table. You cannot hierarchically join two or more tables with non-MOSAIC updating.
- With MOSAIC (Minimum Occurrences of Segments Available In-Core) processing, the application is written as if the entire logical record exists in memory at one time, but VISION:Builder optimizes by only retrieving and updating the segment occurrences required by the application. In fact, only one segment occurrence of each segment type is in memory at the same time, reducing the amount of memory needed to hold the logical record. In general, the use of MOSAIC processing produces the same results as standard processing, but you cannot use variable length (VARCHAR) fields.

MOSAIC processing is permitted for M4OLD and M4CORDn. It is specified by using the MOSAIC keyword on the FILE MASTER or FILE CORDn commands. MOSAIC processing cannot be specified on any input file from which an entire record select subfile is created. When M4OLD is specified with MOSAIC processing, M4NEW is not allowed in the same run. With MOSAIC processing, no distinction is made between the master file qualifiers, N or O, because all qualifiers refer to the same copy of the segment. Own-code and User I/O cannot be used on a file to be accessed with MOSAIC processing.

Aside from the storage savings from having at most one occurrence of a segment type in storage at one time, MOSAIC processing offers I/O optimization. Because a segment is not retrieved until it is referenced, unused segments are not retrieved. This requires that you know the structure of the database. For example, if a segment is referenced in multiple requests, it is retrieved in each request, creating additional I/O. Also, multi-path nested looping could cause unnecessary I/O.

The process of retrieving rows from a relational database to build a logical record is described in Chapter 3, *Defining Data*. As stated, VISION:Builder maps (retrieves) a row from a relational table into a logical segment by applying the rules provided by the logical relation and WH statements. In general, the application of these rules, which are optional for the root segment, produces several rows that satisfy the rules. Consequently, these are mapped into repeated segments of a logical record. Within the logical record, instances of the segments are distinguished by the key fields defined for the segments. When you intend to update a segment, the set of key fields defined should be chosen to establish uniqueness across the segments. Key fields are not an integral part of relational databases so any fields in the segment can be key fields. You can define up to nine key fields in each segment.

Updating a relational table is intuitively what you would expect it to be. A row is retrieved, updated, and returned to the row from which the data was originally retrieved. This rule is obeyed regardless of the changes made to the key fields by the application. You cannot make VISION:Builder retrieve data from one row and, by changing the key fields, put the data back into another row. Updates are performed based on key values and VISION:Builder assumes that the keys uniquely identify a row.

Updates occur during the transaction cycle or within a procedure during the processing cycle. If logical segments are created, new rows are added to the tables represented by the appropriate segments. If logical segments are updated (that is, the contents of the segment changed), the row is replaced in the table. If a logical segment is deleted, that segment and all its subordinates are deleted from the logical record, but only the row representing the logical segment is deleted from the table. It is up to the referential integrity facilities of the relational database manager to determine if any rows in the other tables corresponding to the subordinate segments are affected. Designing for referential integrity is described later in this chapter.

The MOSAIC updating process of a record in a relational database can be conceptualized as follows:

- The root segment is retrieved for a record of the master file.
- Updating, using transaction processing, is performed.

  For each transaction that applies to one leg of this master file record, each individual non-root segment in that leg is accessed directly by key values, moved into the logical record, updated according to the transaction, and written back out to the respective table.
- Procedures are executed and updating is done if needed.

  As each segment is referenced in a procedure, that row is retrieved from the table into the logical record. If changes occur in that segment, it is written to the table before the next occurrence is retrieved.

To return the segments to the tables, invoke the logical relation that was used to retrieve the segments, connected using the AND operator with the values of the key fields as *originally retrieved* from the tables. WH statement rules are not involved in the update. If the retrieved rows are uniquely identified by the keys, this process returns the segments to their original table rows. It is not possible to retrieve a row and modify the keys to direct the row back to update some other row, because VISION:Builder always uses the original key values (not the modified key values) when updating a row.

In this process of retrieval of rows into segments and the subsequent replacement of segments into table rows, the logical relation provides rules for retrieval of rows into the logical record and for writing the rows back to the external relational tables. Their basic format is table column name = VISION:Builder field name. Should any of these values in the logical relation change *between* retrieval and

update, a different row of the table is updated, not the row that was retrieved. This happens because the original values used in the rule for retrieval changed prior to applying the update.

The changes occur due to the VISION:Builder application replacing a different value into the VISION:Builder field name of the logical relation or another concurrent application altering the value of the table column name in the relational table.

To ensure that rows are directed back to the tables properly, any fields involved in the logical relation should not be modified after retrieval and prior to update, because their purpose is identical to that of key fields.

However, if a table column in a logical relation changes as the *result* of an update, the row is directed as expected (to the same row from which it was retrieved) but subsequent retrievals are affected. This is illustrated here to point out that a change in a table column due to an update is allowable but directly affects subsequent retrievals.

Consider an application where you have a logical relation that constrains your view to individuals with salaries less than $26,000.

```
SELECT empno, salary FROM EMPLOYEE_TABLE X1 WHERE salary < 26,000
```

The VISION:Builder procedure brings each relevant row into the logical record as it loops through each segment occurrence. If the application changes the salary of an individual to $29,000, that segment is placed into the table before looping on the next occurrence.

```
UPDATE EMPLOYEE_TABLE X1 SET salary = 29,000
    WHERE empno = 231 AND salary < 26,000
```

In the next procedure, that individual does not satisfy the logical relation and, therefore, does not appear in the logical record.

VISION:Builder optimizes by only updating a row if data changes in the segment. It issues an UPDATE call to the relational database manager if any data changes due to transaction processing or procedural updates, or if the segment contains any invalid fields. An invalid field prints as an asterisk (*) on a report and is described in Chapter 4, *Field Characteristics and Handling*.

By issuing the UPDATE call when invalid fields are involved, VISION:Builder preserves or creates the NULL attribute of a column in a relational table. See Chapter 16, *Database Access Methodology* for detailed information on NULL attribute handling in VISION:Builder.

## Set Operations

There is a consequence of the use of relational tables that requires extra consideration when designing applications that update tables. Operations performed by the relational database manager against a relational database are set operations. Set operations apply to all rows that satisfy the row identification criteria. So if the segments of a logical record are not unique and one of them is

updated, it is equivalent to all of them being updated in the same manner. The update is applied to all rows of the table that satisfy the logical relation connected with the original key values using the AND operator.

Therefore, it is important to identify a group of key fields in the logical record that establishes the uniqueness of the segments. It is not sufficient that the original rows in the table be unique; it must be the subset of fields selected as key fields for the VISION:Builder logical record that are unique.

For example, consider the file definition and mapping specifications in Figure 15-33 and the data in a typical logical record in Figure 15-34.

```
                                    FILE DEFINITION - STAFF

RECORD FORMAT = SQL                                                  FILE IDENTIFICATION =
BUFFER SIZE = 31000                                                  NUMBER OF SEGMENTS IN FILE = 3

                                          +---------+
                                          °DIVISION °
                                          +---------+
                                               °
                                          +---------+
                                          °DEPARTMT °
                                          +---------+
                                               °
                                          +---------+
                                          °EMPLOYES °
                                          +---------+

                                    ********************************
                                    *    SEGMENT   1, LEVEL 1      *
                                    *   SEGMENT NAME = DIVISION    *
                                    ********************************
-------------------------------------------------------------------------------------------------------------------------------
FIELD   FIELD   FIELD    FIELD   FIELD    DEC    CNT FIELD  EDIT CODES    EDIT    OUTPUT   LINE  DATA BASE NAME/
NAME    TYPE   LOCATION  LENGTH  RNDING  PLACES  FOR SGMT   ( ) ( ) ( )  LENGTH   WIDTH     NO   *** COLUMN HEADING ***
-------------------------------------------------------------------------------------------------------------------------------
DEPTNUM   C        11       3                                              3       3          DEPARTMENT_NUMBER
DIVNAME  C/S        1      10                                             10      10          DIVISION_NAME

                                    ********************************
                                    *    SEGMENT   2, LEVEL 2      *
                                    *   SEGMENT NAME = DEPARTMT    *
                                    ********************************
-------------------------------------------------------------------------------------------------------------------------------
FIELD   FIELD   FIELD    FIELD   FIELD    DEC    CNT FIELD  EDIT CODES    EDIT    OUTPUT   LINE  DATA BASE NAME/
NAME    TYPE   LOCATION  LENGTH  RNDING  PLACES  FOR SGMT   ( ) ( ) ( )  LENGTH   WIDTH     NO   *** COLUMN HEADING ***
-------------------------------------------------------------------------------------------------------------------------------
DEPTNAME  C        4       36                                             36      36          DEPARTMENT_NAME
DEPTNO   C/S       1        3                                              3       3          DEPARTMENT_NUMBER

                                    ********************************
                                    *    SEGMENT   3, LEVEL 3      *
                                    *   SEGMENT NAME = EMPLOYES    *
                                    ********************************
-------------------------------------------------------------------------------------------------------------------------------
FIELD   FIELD   FIELD    FIELD   FIELD    DEC    CNT FIELD  EDIT CODES    EDIT    OUTPUT   LINE  DATA BASE NAME/
NAME    TYPE   LOCATION  LENGTH  RNDING  PLACES  FOR SGMT   ( ) ( ) ( )  LENGTH   WIDTH     NO   *** COLUMN HEADING ***
-------------------------------------------------------------------------------------------------------------------------------
CLASS    F/S       1        2                                              7       7          CLASS_NUMBER
SEX       C        3        1                                              1       1          SEX
```

Figure 15-33 File Definition and Mapping Specification – Abbreviated Glossary Listing

```
WIDGETS
       A01 FINANCE
           45    M
           23    F
       E05 LEGAL
       B01 SALES
           23    M
           34    M
           29    F
           34    F
           23    M
       C02 ENGINEERING
           44    M
           44    M
```

Figure 15-34 Logical Record Prior to Updating

In the file definition, CLASS is defined as the key field. As can be seen, the keys are not unique.

If you were to provide the following transaction to update this record, you would retrieve the first row under department B01 that contained 23 in column CLASS and apply the transaction to that segment occurrence.

```
Transaction definition:     Match       Match       Match       Replace
Transaction data:           WIDGETS     B01          23         F
```

Only one transaction is provided and it is applied against *one* logical segment. However, when VISION:Builder instructs the database manager to apply the update to the database, the database manager applies the update against all table rows satisfying the logical relation and having the same key. Because CLASS is the key, all of the rows with the value of CLASS = 23 are located and updated in exactly the same way. The result is that multiple rows are updated with only one transaction and one retrieval operation. The updated record is shown in Figure 15-35.

```
WIDGETS
     A01     FINANCE
             45            M
             23            F
     E05     LEGAL
     B01     SALES
             23            F     <  Retrieved & updated
             34            M
             29            F
             34            F
             23            F     <  Updated
     C02     ENGINEERING
             44            M
             44            M
```

Figure 15-35 Logical Record After Updating

The same logic applies with respect to the deletion of rows from a table. In this example, delete the first entry with a class of 23. The transaction is:

```
Transaction definition:        Match      Match     Delete
Transaction data:              WIDGETS    B01       23
```

This results in both the rows with CLASS = 23 (in the sales department B01) disappearing. On the next retrieval, the logical record available is shown in Figure 15-36.

```
WIDGETS
     A01     FINANCE
             45            M
             23            F
     E05     LEGAL
     B01     SALES
             34            M
             29            F
             34            F
     C02     ENGINEERING
             44            M
             44            M
```

Figure 15-36 Logical Record After Deleting

None of the above is unique to relational database systems. In all forms of database management, if the keys are not unique, the results are rarely acceptable.

Set operations are a consequence of the way that relational database managers work. Although there can be only one occurrence of a segment in a logical record, it is possible to affect more than one row of the corresponding table. This occurs when the Suppress Duplicates specification enforces a unique segment in the logical record or with the root segment, which by definition is unique and is the first row available or the first row satisfying the logical relation. The update or delete process is the same. All rows of the table satisfying the logical relation with the identified keys are affected.

When performing insert operations, VISION:Builder makes every effort to eliminate the possibility of inserting a duplicate row into the relational table. It accomplishes this through the definition of keys in the logical file definition and the match actions required to apply a transaction. A transaction that does an insert must match on all key fields in the segment. VISION:Builder tries to retrieve a row from the table that matches on all the key values and only allows the insert to proceed if no row is found.

It is, therefore, very important to all aspects of updating, inserting, and deleting that the keys be defined to establish uniqueness across the segments of the logical record. VISION:Builder permits the definition of up to nine keys per segment to establish uniqueness.

It is important to note that VISION:Builder can only change fields in the logical record. Once output, those fields can update fields in the physical table. VISION:Builder, therefore, cannot update fields in relational tables that have not been defined in a file definition. In addition, if a column of a table is to be updated, it must be defined as a mapped field in a file definition. A mapped field is one for which a logical field has been defined and a physical field has been specified from which to transfer data (that is, it has an entry for the DB2 Name on the Segment Definition window). An unmapped field is a field defined in the logical record that has no physical counterpart (that is, the field exists only in memory and not in a relational table).

## Referential Integrity

The power of relational tables is derived from the fact that the tables have no implicit relationships between them. All relationships are established explicitly. This is generally achieved by the matching of one or more fields in each of the tables. Those rows that have matching field values satisfy a relationship for an application.

In VISION:Builder, the matching fields are established by means of the logical relation. Therefore, if the value of a field in one table is changed but the value of the corresponding field in the second table is not changed, the original relationship is no longer satisfied. The linkage is broken and the integrity of the database is damaged.

For example, review these sample tables in Figure 15-37.

```
DIVISION_TABLE

 REGION          DIV

 EAST            WIDGETS
 EAST            SPURS
 WEST            LOCKS


 DEPARTMENT_TABLE

 DIVISION         DEPTNO     NAME            MANAGER_NO

 WIDGETS          A01        FINANCE            123
 WIDGETS          E05        LEGAL
 WIDGETS          B01        SALES              456
 WIDGETS          C02        ENGINEERING        789


 EMPLOYEE_TABLE
 EMPNO      NAME      CLASS      SALARY      DEPT      LOC      SEX

 231        JONES      23        25000        B01      L65       M
 234        SMITH      34        45000        B01      L80       M
 123        WILSON     45        75000        A01      L65       M
 477        BROWN      29        35000        B01      L80       F
 456        WHITE      34        45000        B01      L75       F
 333        GREEN      44        25000        C02      L80       M
 789        BLACK      44        67000        C02      L80       M
 567        BLUE       23        25000        A01      L65       F
 444        ADAMS      23        25000        B01      L65       F
```

Figure 15-37 Sample Relational Tables

If you were to change the Sales department number, B01, to a value of H01, in the DEPARTMENT_TABLE, the resulting table would be as shown in Figure 15-38.

```
 DEPARTMENT_TABLE

 DIVISION      DEPTNO      NAME         MANAGER_NO

 WIDGETS        A01        FINANCE         123
 WIDGETS        E05        LEGAL
 WIDGETS        H01        SALES           456
 WIDGETS        C02        ENGINEERING     789
```

Figure 15-38 Department Table with Sales Department Number Changed

Notice that the integrity of the database is destroyed. The EMPLOYEE_TABLE shows nobody in the Sales department (nobody has department code of H01) and there are several employees with a department code of B01 for which there is no department in the DEPARTMENT_TABLE.

If you delete the Sales department from the DEPARTMENT_TABLE, you get the table in Figure 15-39.

```
DEPARTMENT_TABLE

 DIVISION     DEPTNO     NAME         MANAGER_NO

 WIDGETS       A01       FINANCE         123
 WIDGETS       E05       LEGAL
 WIDGETS       C02       ENGINEERING     789
```

Figure 15-39 Department Table with Sales Department Number Deleted

This is just as unacceptable as changing the department number, because you now have no department B01 but have employees in the EMPLOYEE_TABLE with this department code.

Similarly, if you add a new employee with a department code of G34, you have an employee without a department.

In the above example, the employee table serves as a master list of employees and the department table acts as a master list of all valid department numbers. Other tables refer to the same entities described in these tables. When a table contains an entity for which there is a master list, that entity should actually appear in the master list or the reference will be invalid. The table that contains the master list is the parent table and the table that refers to it is a dependent table. The conditions of a set of tables in which all references from one table to another are valid is called referential integrity.

DB2 provides referential integrity within the database manager. Referential integrity is the automatic enforcement of referential constraints. A referential constraint defines a relationship between a parent table and a dependent table. A parent table contains a primary key and a dependent table contains a foreign key. The referential constraint states every value of a foreign key in a dependent table must appear as a value of the primary key in the parent table. This ensures data consistency. The referential constraint is created when the foreign key is defined and it is related to the primary key.

For example, the rule that every department number shown in the sample employee table must appear in the department table is a referential constraint. This ensures that every employee belongs to an existing department. In this case, the department table is a parent table and its column of unique department numbers (DEPTNO) is a primary key; the employee table is a dependent table and the column of employee department assignments (DEPT) is a foreign key. Every value of DEPT must match a value of DEPTNO.

In a VISION:Builder application, the referential integrity of the database manager takes precedence. VISION:Builder does not interfere with it. In fact, by virtue of the logical record, VISION:Builder confirms that some of the referential constraints are

adhered to and ensures that the user does not violate them. In the absence of referential constraints on the relational database manager side, VISION:Builder serves to enforce some form of referential constraints from the application side.

Within relational databases, there are no fields defined as key fields to the database manager. However, for the purposes of an application it is convenient to *think* of certain fields as key fields. Key fields within relational databases are, therefore, a concept rather than an entity. Because key fields are not defined explicitly to the relational database manager, any field can assume the role of a key field. It is purely a function of the application. Because logical key fields (segment keys) have a significance to the operation of VISION:Builder, segment key fields must be defined to VISION:Builder.

## Designing for Referential Integrity

To preserve the integrity of the database, corresponding fields in different tables must be changed at the same time or at least maintained in synchronization. This is not a task that can be achieved automatically by a program. This is a task for the application. The problem is that not only must the keys and foreign keys as they pertain to the current application be synchronized, so must all fields used as keys or foreign keys in all applications. This is not information known to any single program.

Consider the Department-Employee example above. Suppose that you change the MANAGER_NO of the Finance Department (A01) to 888.

- Does this mean that the employee number for WILSON (who used to be the manager) should be changed to 888?
- Should you demand that a new employee with employee number 888 be added to the EMPLOYEE_TABLE or is this an error?

Here you are dealing with a field that is not used as a key or in the logical relation in any of the examples in this document. It is clear, therefore, that the preservation of referential integrity is the responsibility of the application and the relational database manager. You must be concerned not only with the task at hand but with the integrity of the database.

This section will discuss referential integrity in a VISION:Builder application under three headings:

- The maintenance of values of fields in separate tables.
- The insertion of rows in a table.
- The deletion of rows from a table.

### Maintenance of Values of Fields in Separate Tables

The maintenance of values (in fields, foreign keys, and primary keys) is the responsibility of the application. The corresponding referential integrity checks are handled entirely by the relational database manager. The application need not reference all of the tables in the relationship for the checks to occur according to the

referential constraints. For example, consider an application that transfers an employee from department B01 to E05 by changing the department number in the EMPLOYEE_TABLE and does not reference the DEPARTMENT_TABLE. The relational database manager ensures that department E05 exists in the DEPARTMENT_TABLE even though the application never used that table.

However, when the values of fields in rows of one table are updated, you must consider what other fields in what other tables need to be updated at the same time. VISION:Builder provides simple straightforward capabilities for updating fields in an application.

To preserve referential integrity, make sure that all of the fields to be maintained are defined and updated in the segments that represent the tables. VISION:Builder can only update those physical fields in the tables for which it has a logical representation in a segment.

In the examples, you have not used the field MANAGER_NO in the DEPARTMENT_TABLE. It is reasonable to assume that a person can only be a manager if that person is actually a member of the department. Assume that MGRNO is the representation of MANAGER_NO in the DPRTMNT segment. In transferring a manager (that is, changing the department number in the employee table), you may not know who the new manager is, but you certainly do not want to leave the incorrect manager in the department table. So, blank out the manager number if the manager ceases to be in the department.

The file structure and contents are shown in Figure 15-40.

```
File: STAFF

Segments     Fields                                          TABLES
divisn        divname, region                                DIVISION_TABLE
  deprtmnt     dptno, deptname, mgrno                        DEPARTMENT_TABLE
    emps        empnum, dept, name, class, sal, locn         EMPLOYEE_TABLE
```

Figure 15-40 File Structure for STAFF

The relevant update is achieved by the following three simple statements:

```
DO WHILE (empnum EQ mgrno) AND (dptno NE dept)
                          LET mgrno = ' `
                               END
```

## Insertion of Rows in a Table

When inserting a row into a table with referential constraints, the database manager makes sure that a primary key in the inserted row is unique and that a foreign key contains a value that exists in the parent table.

The structured logical record greatly assists in preserving referential integrity with the insertion of rows into a table because the mechanics of the VISION:Builder transaction insertion capability demand the existence of parent segments and the uniqueness of the inserted segment occurrence. You cannot insert a segment without the appropriate parent segments existing. These parent segments must

either exist or be inserted in the same transaction. Because parent segments in a logical record are other related tables, the VISION:Builder process ensures rows exist or are inserted into all related tables.

For example, assume that you had a logical structure of:

```
Customer
   |
 Order
   |
 Item
```

It is not possible to define a series of actions to insert an item without having a customer and an order.

The logical record, merely by its existence, provides you with a means of controlling referential integrity when rows are inserted into a relational table.

VISION:Builder capabilities control tables not used in the application and assist in preserving the integrity of the entire database and not just the relational tables used in the application. For example, review the file structure of the logical record in Figure 15-40. The insertion of a new employee demands that a division and department exist. Suppose that there is also a table that records the employees retirement plan contributions (RETIRE_TABLE) and another table that records the vacation hours (VAC_TABLE). As an employee is added to the EMPLOYEE_TABLE, add a row to each of the other two tables. To do this, include two logical segments that represent the tables in the logical record, each subordinate to the employee segment.

The structure is shown in Figure 15-41.

```
File: STAFF

Segments     Fields                                                TABLES
divisn        divname, region                                      DIVISION_TABLE
 deprtmnt      dptno, deptname                                     DEPARTMENT_TABLE
   emps          empnum, name, dept, class, sal, locn              EMPLOYEE_ TABLE
    retire         empno                                          RETIRE_TABLE
    vac            empnumb                                        VAC_TABLE
```

Figure 15-41 File Structure for STAFF with Segments Added for Retirement and Vacation

The application ensures that whenever an insert is made for the EMPS segment, inserts are also made for the RETIRE and VAC segments. The net result is to update three tables with a single transaction. For example, when inserting a new employee number 7, the transaction is:

```
Transaction definition:  Match     Match    Insert     Insert     Insert
Transaction data:        WIDGETS    B01       7          7          7
```

EMPNUM and DEPT must be defined in the EMPS segment so that the values for DEPT are inserted into the logical record and into the new row of EMPS table. The inserted segment creates a new row on the table with data provided for the following fields:

```
Logical Field        Physical
  empnum              EMPNO
  dept                DEPT
  name                NAME
  class               CLASS
  sal                 SALARY
  locn                LOC
```

The column SEX, which exists in the table, cannot be updated by the insert action because it has no representation in the logical segment. This field is set to the value of NULL by the database manager if the column was defined as NULL. If, when the CREATE TABLE was established, the column SEX was defined to be NOT NULL, when VISION:Builder issues the INSERT call, the database manager rejects the insert and returns an error code to VISION:Builder. If it was defined as NOT NULL WITH DEFAULT, the insert places the appropriate default value into the column and inserts the row.

In the previous example, you were concerned with the referential integrity of the database and wanted to make sure that whenever a row was inserted in one table, corresponding rows were inserted into other tables. You achieved this by use of the logical record capability and following the VISION:Builder constraints upon a hierarchical structure. But suppose that you wanted to insert a lower level segment and did not want to update the parent segments or even care if they existed. For example, in the following structure you can well want to insert the ITEM segment without concern for the customer or order segments.

```
customer
|
order
|
item
```

The truth is that you should not be using this application to insert rows into the ITEM table. You must use a different set of logical records isolating only the ITEM table and performing the updates on the ITEM table only. The ITEM table is defined in the application as the master file. To reference the customer and order information as well, define it as the coordinated file. The set of logical records would then be:

```
item     customer
         |
         order
```

This demands information about the coordinated file (customer and order) be supplied on the Files Used window to establish coordination between the logical files. With this structure, inserts can be performed on the master file record (item) without affecting the coordinated files (customer and order).

VISION:Builder support of relational databases does not constrain you to work with a specific set of structures as dictated by the database manager. Using relational databases, you can select the structures that support the existing

referential integrity. Each of the tables can be handled separately without the use of the logical record capability. In this case, processing considerations are similar to those required of a COBOL application. You gain control at the expense of the automatic functions provided by the logical record. On the other hand, either a single logical record or multiple logical records can be used. It all depends on what you want to control and what you want VISION:Builder to do for you.

## Deletion of Rows from a Table

The VISION:Builder rules for the deletion of a segment automatically cause the deletion of the segment, plus all subordinate segments in the logical record. However, a DELETE call to the database manager is issued only for the specified segment. VISION:Builder does not issue DELETE calls for any of the subordinate segments. When performing the physical delete for the row on the database side, it can well be that it should be deleted from several other tables or it can be that deleting a row from a table should cause foreign keys in other tables to merely be set to null. It is up to the relational database manager to determine the effects of a delete; VISION:Builder does not interfere with that process. Using the example above, where you wanted to insert into several tables, you can assume that you also wanted to delete from several tables. You have the logical structure:

```
File: STAFF
Segments                        TABLES
divisn                          DIVISION_TABLE
     deprtmnt                   DEPARTMENT_TABLE
          emps                  EMPLOYEE_TABLE
            retire              RETIRE_TABLE
            vac                 VAC_TABLE
```

The deletion of an employee by a single transaction to delete the EMPS occurrence deletes the relevant EMPS occurrence and the subordinate RETIRE and VAC segments in the logical record (each of which occurs only once in this example). However, VISION:Builder issues only one DELETE call for the EMPS occurrence to the relational database manager. The referential constraints determine whether or not the rows are deleted from the RETIRE and VAC tables.

The delete action is also a set operation and if the keys are not uniquely established, the delete action removes all rows that satisfy the logical relation and have the same key values.

While the logical record greatly assists in the preservation of referential integrity when rows are deleted from tables, there are situations where you will not want to cascade the delete to lower level segments.

Consider again the logical record structure:

```
customer
|
order
|
item
```

Here, in the course of the deletion of a CUSTOMER record or, perhaps more realistically an ORDER record, you certainly do not want to delete the items from the ITEM table. As with the update situation, the answer lies in selecting the logical

records that provide support for the application. In the case cited, you may want to have the following structures with a master file (customer and order) and a coordinated file (item).

```
customer      item
|
order
```

Once again, this requires information about the coordinated file (item) be supplied on the Files Used window. If wanted, the customer and order tables are handled separately. In this case, deletions against one record do not affect the other records. As before, the effects that a delete has on the physical tables are determined by the relational database manager. Your choice of logical record structure in an application reflects the table relationships. In this case, the delete operates on the logical and physical entities in a similar fashion. However, it is not a requirement.

## The Order of Updating Rows

In this section, the term update is used in its broadest sense to mean any change to the table. Update, in this section, includes the functions of UPDATE (change to data in rows), DELETE (removal of rows), and INSERT (creation of rows).

The logical record concept used within VISION:Builder involves mapping records from relational tables into logical in-memory records, applying transactions, and returning the updated rows to the original table. Rows are returned to their original position by the use of the logical relation used for retrieval connected with the VISION:Builder key fields using the AND operator. By using the original data for the key values, you ensure that the same rows are located.

The above process assumes that none of the tables or VISION:Builder fields involved in the logical relation are changed between retrieving the rows and returning them. If any of the values of the keys or foreign keys in the rows of the tables are changed during the execution of an application or any values in the logical relation change, it follows that VISION:Builder cannot return the data to the original rows. Fields in the logical relation can be changed by the application. Keys and foreign key fields (columns) can be changed by actually changing the values in the relational table or by deletion of the rows. Tables can be changed between retrieval and update in two ways:

- Another application changes the table.
- A table is represented by different segments at different levels in the same logical view.

The first case can be prevented by invoking the appropriate relational database locking facility. The latter case is an application design consideration.

These situations can be caused by the fact that as VISION:Builder returns the updated segment to the relational table, it does it one segment at a time. After each segment is returned, the table is now modified for the subsequent segments. This means that if a row is updated using two separate segments, the second update is impacted by the results of the first update. If the first update does not affect keys or foreign keys, the second update overrides the results of the first update. This is

exactly the same as providing two transactions against a conventional file. If, however, the first update changes the keys or foreign keys, the second update may not find its original table row. The database will be considered inconsistent and the transaction may be rejected with the appropriate database rollback.

You can easily avoid problems of this nature by ensuring that updates to a table occur using one segment representing the table. When a table is represented by more than one segment in an application, all of the updates should be made against only one of these segment types.

When updates occur on a series of segments in a particular leg or path of the hierarchy, the order in which VISION:Builder makes the individual calls to the relational database manager for each segment is important. When a segment is deleted from a leg of the hierarchy, only one DELETE call is done for the segment. Calls are not made for any subordinate segments. However, they are removed from the logical record in memory.

When a segment (or series of segments) is inserted into a leg of the hierarchy using a transaction, the INSERT calls are made in a top-down order. The INSERT call for the parent segment is made prior to making the INSERT call for the child segment. This order is dictated by the requirements imposed by VISION:Builder when inserting segments with transaction processing.

When a series of segments in a hierarchical leg are updated (that is, the data has changed), the UPDATE calls are made in a bottom-up order. The UPDATE call for the lowest level child segment that has changed is made prior to making the UPDATE call for the parent segment. Recall that the UPDATE calls are conditional and are made only if the data has changed or there are invalid fields in the segment.

Chapter

# 16 Database Access Methodology

The information in this chapter is directed towards database administrators and application programmers concerned with the control, tuning, and optimization of their database access in a VISION:Builder application.

This chapter is organized into sections, each relating to an individual database manager or access method. Each section discusses various topics relevant to the interaction between VISION:Builder and the particular database manager or access method. It requires you to be familiar with the database manager and the types of calls necessary to retrieve data from the database.

Each section discusses topics such as the types of calls that VISION:Builder issues to the database manager, how an application can affect the types of calls issued, performance considerations, and storage utilization issues.

## IMS Processing

The VISION:Builder IMS interface makes the full range of VISION:Builder capabilities available for processing IMS databases. It offers complete transparency; you do not need to know that an IMS database is being processed when writing your procedures. An IMS database is processed like any VISION:Builder structured file. See Chapter 3, *Defining Data* for defining IMS databases to VISION:Builder.

All considerations for structured records (such as looping, multi-path nested, and parallel looping) also apply to IMS records. Two features of the interface, memory optimized (MOSAIC) processing and preselection requests, are available to optimize performance and are described later in this chapter.

As far as IMS is concerned, VISION:Builder is a batch application program. The interaction between IMS and VISION:Builder is the same as between IMS and an application written in any language. Refer to the *VISION:Builder for OS/390 Environment Guide* (OS/390 users) or the *VISION:Builder for VSE Environment Guide* (VSE users) for specification of the JCL to execute VISION:Builder as an application under IMS.

In execution, VISION:Builder is under the control of the IMS region controller. All access to the databases is requested through calls to the IMS modules. These CALL statements are automatically generated internally by VISION:Builder. Because IMS is performing all I/O to the databases, the DTF/DD names used to refer to the databases must be the ones known to IMS (that is, the ones coded on the DATASET statement of the DBD). The standard VISION:Builder DTF/DD names for files M4OLD, M4NEW, M4CORDn, or M4SUBFn should not be coded in the JCL if that file is a database.

When IMS passes control to VISION:Builder, it provides a parameter list containing the PCBs available for the run and issues a message indicating its selection. Before beginning processing, VISION:Builder selects the appropriate PCB to be used for each IMS file in the run and issues a message indicating its selection. The selection is based on the processing option (PROCOPT) for the appropriate DBD. VISION:Builder uses the PCB with that file's most restrictive PROCOPT.

For a retrieval only file, a PCB with a PROCOPT=G will be selected over one with a PROCOPT=A. VISION:Builder only checks the first character of the PROCOPT field and the only valid options that it recognizes are G, A, and L. That means that if a PCB has a PROCOPT=RG, it is rejected as a retrieval PCB, while one with a PROCOPT=GR is accepted. If two PCBs for the same DBD have the same PROCOPT, the first one will be used. When more than one database is used in the run, a separate PCB is required for each database. PCBs are not shared by VISION:Builder.

PCBs are selected in a specific order based on the VISION:Builder file designation. The selection order is M4OLD, M4NEW,❹ M4CORD1 through M4CORD9, M4SUBF1 through M4SUBF9, then M4SUBF0. The order and selection method can be very important when the same database is being used as more than one file in your application (such as M4OLD and M4CORD1). Because a separate PCB is selected for each file, the order of the PCBs within the PSB may be important so that the PCB with the appropriate processing options (PROCOPT) is positioned correctly for selection by VISION:Builder.

For M4CORDn and M4OLD, when no transactions or update-in-place are specified,❹ the most restrictive processing option is G. If transaction and/or update-in-place are specified,❹ M4OLD must have a PROCOPT=A. M4NEW❹ and M4SUBFn must have a PROCOPT=L or PROCOPT=LS as appropriate to the database organization.

M4OLD and M4NEW❹ cannot be specified in the same run unless the database organization is HSAM or SHSAM. See IBM's IMS System/Application Design Guide for a discussion of HSAM/SHSAM processing. Note that IMS input files (master file and coordinated files) must be listed on the FILE command in the Run Control command group.

## Standard Processing

### Retrieval

The default method of processing is standard processing. This involves VISION:Builder retrieving an entire logical record (all occurrences of all sensitive segments for a given root) into the appropriate work area(s) (M4OLD, M4NEW,❹ M4CORDn). All processing activity is performed against the data in the work area. No further retrieval access to the database is done.

The IMS CALL statements generated by VISION:Builder are to retrieve the root segment according to the type of processing and retrieve all sensitive dependent segments with unqualified GNP calls. For straight sequential retrieval (of M4OLD or M4CORDn), a get next (GN) call with an unqualified SSA is used to retrieve the root. If start search is used, M4OLD is accessed with a get unique (GU) call with an SSA qualified as (ROOTKEY GE start value).

For indexed coordination or user coordination, the call to the root is a get unique (GU) with a fully qualified SSA (ROOTKEY=value). See Chapter 6, *Input File Processing* for information about coordination.

With standard processing, IMS only retrieves a segment one time for processing. Referencing a segment in multiple requests or in a loop has no impact on I/O.

For this reason, standard processing should be used in jobs that require a given segment to be referenced in multiple requests. It also offers minimal impact on the system by users who are not familiar with the database structure. It should also be used in generalized update jobs. Standard processing does, however, require a large enough VISION:Builder work area to hold an entire logical record. This space requirement can be minimized by using a PCB sensitive only to the segments used in the run.

When sequentially processing HDAM databases, it is important to remember that the root segments are not in key sequence; they are in random order. VISION:Builder always processes an HDAM database as unsequenced. Any sequence checking specification is ignored. The use of STARTKEY/ENDKEY search values on the FILE MASTER command are ignored and a warning message is issued. A direct-read master file, indexed coordination, and user coordination are well suited for use on HDAM databases. Refer to IBM's IMS Application Programmer's Reference Manual for details on how IMS processes CALL statements with qualified SSAs.

### Updating❹

The only file that can be updated in VISION:Builder is the master file. This discussion on updating an IMS database applies only to the master file. Updating is only possible with the appropriate level of the database interface option of VISION:Builder. See Chapter 10, *Transactions* for information about updating the master file.

In retrieval only runs, VISION:Builder can process unkeyed and non-uniquely keyed segments. For update runs, however, the segments must all be uniquely keyed and segment ordering specified.

If the database is being created as M4NEW, all segments are inserted. VISION:Builder issues an insert (ISRT) call with an unqualified SSA at the appropriate level. Updates done through transaction processing or in a procedure are held in the M4NEW work area until all processing against that record is completed.

If the database is being updated-in-place, the segments in the M4NEW work area are compared with those in the M4OLD work area. Any difference in the values of the key fields results in the appropriate update action (insert (ISRT) or delete (DLET) call) being performed. If the key fields have not changed and only the data has changed, a replace (REPL) call is performed. When update-in-place is requested, VISION:Builder issues a get hold unique (GHU) with fully qualified SSAs before issuing delete (DLET) or replace (REPL) calls. No attempt is made to match (with a get hold call) a segment in the new work area with the database until the entire record is ready for update. VISION:Builder processes a changed key value as a delete and insert, rather than a replace.

You should also be aware that although the PCB allows updates, some updates may not be permitted for IMS logical databases. IMS makes the processing of its logical database structure composed of segments of more than one physical database transparent to an application program. However, the RULES parameter in the DBD may restrict update processing of some segments in a given IMS logical database structure, regardless of the PROCOPT coded in the PCB.

If update action is not permitted in the IMS logical database, IMS returns a non-blank status code to VISION:Builder and VISION:Builder terminates with an unexpected status code message.

### Subfiles

You can direct your subfile output to an IMS database with the appropriate level of the database interface option of VISION:Builder (see Chapter 7, *Output File Processing* for more information about subfiles). You can only select entire records when output is directed to an IMS database. The input file that is selected to the subfile can be an IMS database itself or a non-IMS file format. The NAME entry on the FILE SUBFn command for the subfile, which matches the subfile-name on the EXTRACT FILE command, must be the IMS DBD name for the database being created. You can also use the EXTRACT DBDNAME command to specify the IMS DBD, and then no FILE SUBFn is needed. The file definition used for the input file that is being selected to the subfile must contain segment names and sequence specifications that conform to the IMS DBD for the subfile.

Because the output subfile is processed in write mode, this is an IMS load run (that is, a new database is being created). The PROCOPT in the subfile's PCB must be L or LS as appropriate.

You can also output an entire record from an IMS input file to a non-IMS database. When creating this type of subfile record, segment count fields are added to the end of each parent segment. The count fields are 4-byte fixed binary format. If you request generation of automatic file definition statements for your output subfile, statements are included to define the added segment count fields. They are named in the format QssnQQQQ, where Q is a literal, the ss is a two-digit segment number, and n is the level number of the segment being counted. When segment count fields are added, the output subfile record will not exactly match the input file record. The file definition of the input file will not map the output file.

## Memory Optimized (MOSAIC) Processing

While standard processing brings the entire logical record into the VISION:Builder work areas, memory optimized (MOSAIC) processing brings at most one occurrence of each segment into storage at any one time. A segment is not retrieved until it is referenced, either explicitly or implicitly in VISION:Builder processing.

MOSAIC (Minimum Occurrences of Segments Available In-Core) processing is permitted for M4OLD and M4CORDn. It is specified by entering MOSAIC on the FILE MASTER or FILE CORDn command. MOSAIC processing cannot be specified on any input file from which an entire record select subfile will be created. When M4OLD is specified with MOSAIC processing, M4NEW is not allowed in the same run. With MOSAIC processing, no distinction is made between the master file qualifiers, N or O, because all qualifiers refer to the same copy of the segment. Own-code and User I/O cannot be used on a file to be accessed with MOSAIC processing. Also, the database to be processed with MOSAIC processing must not be flat (root only).

Aside from the storage savings from having at most one occurrence of a segment type in storage at one time, MOSAIC processing offers I/O optimization. Because a segment is not retrieved until it is referenced, unused segments are not retrieved. This means that you must know the structure of the database. For example, if a segment is referenced in multiple procedures of the same type (not subroutines), it is retrieved in each procedure, creating additional I/O. Also, multi-path nested looping could cause unnecessary I/O.

With MOSAIC processing and IMS, the VISION:Builder file definition and the PCB must have a one-to-one correspondence (unlike standard processing that allows the file definition to contain more segments than the PCB). A file definition and PCB that describe more than one hierarchical leg of a database are processed with full MOSAIC processing when MOSAIC processing is specified. On the other hand, if the file definition and PCB contain only one hierarchical leg, single leg MOSAIC processing is used.

Full MOSAIC processing has the requirement that the segments be ordered. It uses get next (GN) calls on lower level segments that are qualified with the SEGKEY GT last key value. In single leg processing, the get next calls are unqualified. This means that single leg processing is the only way (without preselection requests) to process unkeyed or non-uniquely keyed segments with MOSAIC processing. The use of full MOSAIC processing also requires that multiple positioning (POS=M) be coded in the PCB.

Writing applications to effectively utilize MOSAIC processing capabilities requires a knowledge of the database in which the fields are located and the hierarchical dependency of segments. Techniques such as multi-path parallel looping, subroutines, and the FIND function should be used to minimize looping. Otherwise, there may be a degradation in MOSAIC processing as compared to standard processing.

In MOSAIC processing retrieval jobs, VISION:Builder produces the message MK4W212 that follows the standard wrap-up statistics. The MK4W212 statistics reflect the count of segments retrieved to satisfy the processing and reporting requirements. The counts can be used as a testing aid in determining the CALL statements generated by VISION:Builder. Note that, as a minimum, the root segment is always retrieved for each record that is input in the run.

## Updating❹

Updating can be performed with MOSAIC processing. In transaction processing, VISION:Builder uses path calls to match the transaction down a leg. This requires that the PCB have a PROCOPT=AP coded. VISION:Builder also uses a path insert for a transaction that creates more than one segment along a leg.

Updating can also be done in procedures. Replacing data into master file fields results in a segment being replaced. A procedure can dynamically delete a record during transaction processing using the DELETE flag, but cannot dynamically delete segments during transaction processing using the DELETE flag.

With MOSAIC processing and update-in-place, a segment is updated on the database before another occurrence of that segment or its parents are retrieved. This means that a database is updated sooner in MOSAIC processing than in standard processing. A segment or record created in transaction processing with MOSAIC processing is available for coordination (if the same database is being used as both the master and coordinated file). This is not the case in standard processing.

In standard processing, no attempt is made to match (with a get hold call) a segment in the new work area with the database until the entire record is ready for update. VISION:Builder handles a changed key as a delete and insert. However, in MOSAIC processing, an update action cannot alter a segment key. This is because the segment is retrieved with a get hold call and, before another occurrence is retrieved, it is replaced with a REPL call. If the key of the segment has been changed, IMS responds with an unexpected status code of DX and VISION:Builder terminates processing.

If updates are performed in MOSAIC processing, VISION:Builder statistics message MK4W212 and messages MK4W225 and MK4W226 are produced. The MK4W212 message reflects the count of segments retrieved to satisfy the processing and reporting requirements. The MK4W225 and MK4W226 messages detail the total number of successful and unsuccessful CALL statements by type and segment, including path CALL statements.

These statistics reflect all CALL statements in the run: the calls to satisfy the transaction processing requirements as indicated by the transaction definition(s), the calls to satisfy the processing and reporting requirements, and any backout calls that VISION:Builder makes in the event of an unexpected IMS status code causing termination.

There are a variety of techniques used in VISION:Builder for temporarily altering the master work area in one procedure for reporting in subsequent procedures without updating the database itself (for example, replacing values into the new master work area without update-in-place specified or creating and updating a segment through transaction processing without update-in-place). These techniques are suited to MOSAIC processing. By not specifying update-in-place, the database remains unchanged. Temporary updates in the master work area are lost when the next occurrence of that segment type is retrieved. A subsequent procedure receives a freshly read copy of the segment from the database. Likewise, if a segment is created by one transaction, a subsequent match transaction record will not be able to locate that segment on the database.

In general, MOSAIC processing offers not only a substantial main storage savings, but also the possibility of optimizing database accesses. That same capability of optimizing access also makes MOSAIC processing best suited for specific (rather than generalized) reporting and updating. MOSAIC processing is definitely not suited to batching of procedures that process fields in the same segments.

The following example illustrates the difference between using standard processing and MOSAIC processing. Consider the structured record shown in Figure 16-1. It consists of a root segment and two level 2 segments (B and C), each having 10 occurrences.

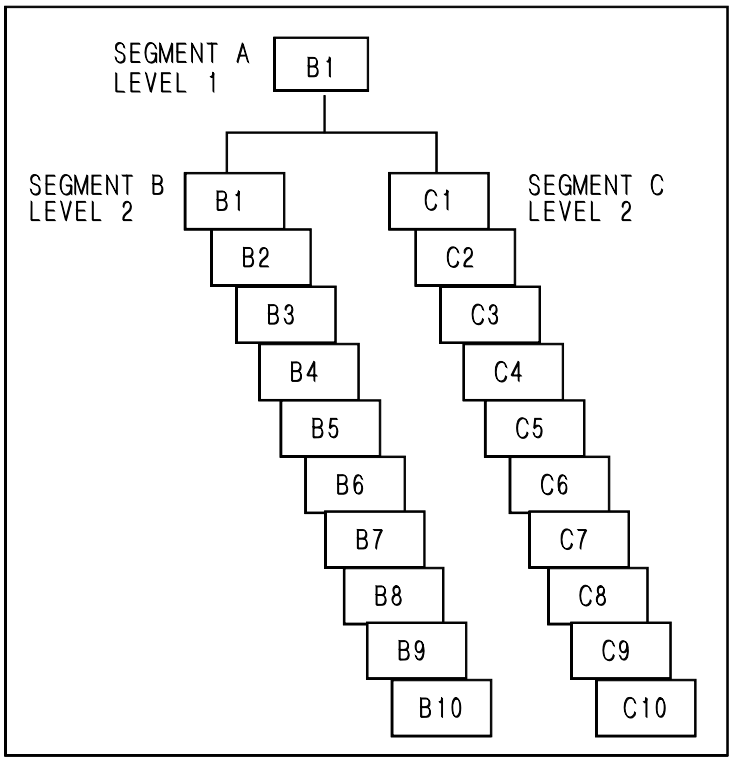



Figure 16-1 Sample Structured Record

Under standard processing, VISION:Builder always brings the entire logical record into storage, regardless of which data fields are referenced, how many times a segment is referenced, or the type of looping required by the application. This requires 21 I/O accesses (one for A, ten each for B and C). This is useful if the data from most segments is to be used.

Under MOSAIC processing, the number of accesses varies with the application. Using the same structure in Figure 16-1, the following three examples show the number of database accesses required when using MOSAIC processing in a single procedure under the following conditions:

- Example 1: Multipath nested looping.
- Example 2: Singlepath nested looping.
- Example 3: Qualified FIND operation.

## Example 1 MOSAIC Processing with Multi-Path Nested Looping

Under MOSAIC processing, if a single procedure requires occurrence B10 and C10, as many as 101 I/O accesses are performed by VISION:Builder. This is because multi-path nested looping causes ten segment C accesses for each segment B access. MOSAIC processing sets up space for only one occurrence of each segment type at a time (total of three spaces). Each new occurrence of segment B that is

accessed requires all ten occurrences of segment C to be retrieved again because they are not retained in memory. I/O is high when MOSAIC processing is used and multi-path loops occur.

### Example 2 MOSAIC Processing with Single-Path Nested Looping

Under MOSAIC processing, if a procedure references segment B10, only 11 I/O accesses are made (one for segment A and ten for segment B). Both time and space are saved.

### Example 3 MOSAIC Processing with a FIND Operation

Under MOSAIC processing, if a procedure uses a qualified FIND function on segment B with key B1, two I/O accesses are required (one for segment A and one for segment B).

## Preselection

Preselection causes a qualified, Boolean SSA to be generated from the information in the SSA expression. This SSA is used during the retrieval of a segment and, as such, acts as a filter on the segments returned to VISION:Builder. This capability can be used to restrict access to particular segment occurrences or to enhance performance by returning to VISION:Builder only those segment occurrences actually needed by the application for both MOSAIC processing and standard processing. In addition, smaller logical records can be built for standard processing.

Preselection is specified using the SEGMENT and SSA operand pairs on the FILE MASTER or FILE CORDn commands (see the *VISION:Builder ASL Reference Guide* for more information). This causes selection logic to be performed on a segment. This logic is transformed into a Boolean SSA by VISION:Builder. Because one SEGMENT / SSA operand pair builds an SSA for one segment only, only temporary fields, constants, and fields from that segment can be referenced in the SSA expression. The database fields must correspond to key or search fields in the DBD. Multiple SEGMENT / SSA operand pairs for the same segment type are not allowed. However, multiple SEGMENT / SSA operand pairs that each reference a different segment type are allowed.

A SEGMENT / SSA operand pair can have up to eight relationships that compare database fields with constants or temporary fields. The constants or temporary fields can have partial field specifications (except for dynamic partial fielding). Normally, VISION:Builder converts two comparison fields to a common data type. The temporary field or constant is converted so that it has the same attributes as the database field.

In other words, VISION:Builder replace conversion rules are used. See Chapter 4, *Field Characteristics and Handling*. The converted value is passed to IMS in an SSA. IMS searches for database field values in the database that exactly match this value in the SSA without doing further conversions. Note that when SEGMENT / SSA operand pairs reference segments in the master file, only the old work area is available. This means that the O qualifier must be used with the database field name.

Inconsistency in the selection logic is not diagnosed by either VISION:Builder or IMS. IMS never returns any occurrences of that segment. Likewise, an invalid field in a SEGMENT / SSA operand pair results in no occurrences of that segment being available. Also, preselection can be used with all or any segment types on either the master or coordinated files.

Preselection cannot be used on the root segment of a file to be read directly (that is, either M4OLD or an indexed coordinated file). It is not allowed on M4OLD when STARTKEY or ENDKEY keywords are specified. Preselection on the root segments of serially read HDAM databases may also cause unexpected results. IMS positions itself as far into a database as possible on the first call, using the lowest value of the key in the SSA.

Because the root segments of an HDAM database are not necessarily in root key sequence, this can lead to bypassing some of the records that fit the qualification. Refer to IBM's IMS Application Programmer Reference Manual for further details on how IMS handles Boolean SSAs. Note that when preselection is used on a lower level segment, the root will always be available. In standard processing, the use of preselection procedures requires that the PCB must be sensitive to all segments in the file definition.

Preselection can be used with MOSAIC processing with care. The PCB must be sensitive to all segments referenced in the SEGMENT / SSA operand pairs and to all segments above them in the same leg of the structure.

When MOSAIC processing and preselection are used together, the manner in which some functions are performed by one feature may be altered. For example, with MOSAIC processing, if the FIND function is used for a segment without preselection, VISION:Builder generates a CALL statement with a qualified SSA, retrieving only that segment. With preselection and MOSAIC processing specified for that segment, the FIND operates interpretively depending on the segments available through the preselection SSA.

Preselection is not allowed if a transaction file❹ is input to the run.

## VISION:Builder Processing and Secondary Indexing

A secondary index provides entry to a database using a field other than the primary root key. This alternate entry point may be in the root or a dependent segment. The entry point may be in one segment (the target segment), while the key may come from a dependent segment (the source segment).

Secondary indexing, therefore, provides the capability of inverting the actual structure when the target is not the root. See Chapter 3, *Defining Data* for information about defining a file structure to VISION:Builder. Once the secondary index structure is properly defined, all types of keyed and non-keyed processing are available using the database as either a master file or a coordinated file.

In cases where transaction processing is used against the secondary structure, the transaction match field should be a character string formatted exactly like the concatenation specified in the SRCH keyword of the XDFLD statement. Due to

IMS restrictions, create, insert, and delete actions are not permitted for the index target segment or any segments on which it is dependent in the original structure. The sequence fields of these segments must not be changed.

## Storage Utilization

The FREESIZE keyword of the CONTROL command in the Run Control command group allows OS/390 and VSE users to specify the amount of storage to be released by VISION:Builder to IMS immediately after the files are opened but before processing begins. This facility allows an application run that accesses a database to execute in a smaller region than would normally be possible.

For example, VISION:Builder normally frees five-eighths of available storage for IMS use during processing. By using the FREESIZE keyword, the amount of freed storage can be reduced below the five-eighths default and the region size can be lowered proportionately.

## Technical Considerations and Hints

There are a number of occasions when neither standard processing nor MOSAIC processing are satisfactory for processing a database. This can occur when MOSAIC processing is the optimal choice but the database contains unkeyed segments on different paths (that is, single leg MOSAIC processing cannot be used). In such cases, it is possible to split the database into a master and coordinated file(s). Defining the various legs according to use (one with single leg, another with full MOSAIC processing, and possibly another with standard processing) and using indexed coordination will often get around the problem. When using this technique, it is important to know which PCB in the PSB VISION:Builder uses for each file.

IMS offers key sensitivity, an option coded in the PCB, as a security feature. When this option is used, nothing is returned in the I/O area; only the key is returned in the PCB key feedback area. In many cases, VISION:Builder cannot handle a segment with key sensitivity. Under some circumstances, it is possible to force VISION:Builder to build a node segment and allow access to the dependents of the key sensitive segment.

In standard processing, retrieval only jobs, the key sensitive segment can be defined as a minimum length (for example, 1 byte). This 1 byte is also defined as the key. When VISION:Builder retrieves a segment, it builds the record from the I/O area, never testing to see if the data has actually changed. In this manner, a node segment is built, allowing access to its dependents, but not actually containing any data.

When standard processing is used, the VISION:Builder work area must be large enough to hold an entire logical record. With IMS databases, this could require considerable storage, especially in cases where MOSAIC processing is not available. The amount of storage required can be somewhat reduced by defining some segments to VISION:Builder as shorter than they actually are. IMS returns the entire segment to the I/O area and VISION:Builder can use only part of the actual segment to build its segment.

In using this technique, it is important to have a segment defined to VISION:Builder that is at least as long as the longest IMS segment, because this determines the length of the I/O area. The selection of segments to be defined as shorter than they actually are must be based on the fields needed by the program. This technique can be used for updates, provided that the short segments are not being updated and that they are long enough to contain the IMS key field.

# Relational Support for DB2

The Relational Support feature of VISION:Builder makes the full range of VISION:Builder capabilities available for processing relational databases under DB2. All interaction with DB2 is handled automatically by VISION:Builder and is completely transparent to you. You develop your application as you would normally for any VISION:Builder structured file. All the powerful features of VISION:Builder are unchanged.

There is no need to know whether relational databases are being accessed to build the record. All considerations for looping in structured records, accessing segments directly by key, and so forth also apply when processing relational databases.

You do not need to learn DB2 (the language used to access data in a relational database). Although, as shown in Chapter 3, *Defining Data*, if you know DB2, you can employ it to improve performance or extend the scope of an application.

## Setting Up and Running a Job for DB2

The use of relational databases requires a new technique for an application whether it is generated by VISION:Builder or COBOL. An application must be bound together with a plan that describes the relational tables and relational activity to DB2. The plan for VISION:Builder is actually created as part of the VISION:Builder installation process at which time the BIND operation is also performed.

You do not have to worry about this other than to ensure that you are authorized to use the plan (or are the original creator of the VISION:Builder plan). All users must be authorized by the DB2 system administrator to use the appropriate plan and relevant tables. Otherwise, DB2 prohibits access to the tables.

VISION:Builder communicates with DB2 using any of three attach facilities provided with DB2:

- TSO Attach Facility
- IMS Attach Facility
- Call Attach Facility

The choice of which facility to use depends upon the requirements of your particular application and your installations standards. Refer to the *VISION:Builder for OS/390 Environment Guide* (OS/390 users) or the *VISION:Builder for VSE Environment Guide* (VSE users) for information about how to use these facilities. For each VISION:Builder application, you can decide which facility suits your needs.

When executing an application using the Call Attach Facility, VISION:Builder connects to DB2 during the compile phase. The connection has already been made for the other Attach Facilities so it is not necessary for VISION:Builder to perform the connect. VISION:Builder remains connected to DB2 for the duration of the application, then closes the thread and disconnects from DB2.

An application that uses the Call Attach Facility must specify the DB2 subsystem ID and the plan name. These are entered on the CONTROL command following the DB2 keyword. See the *VISION:Builder ASL Reference Guide* for more information.

You do not need to supply any additional JCL statements for files in an application that uses relational databases (M4OLD, M4CORDn, or M4SUBFn). The DB2 plan and the file definitions provide the necessary information for accessing a relational database.

## How VISION:Builder Constructs DB2 Statements

During the execution of an application, all access to the relational databases is requested through calls to DB2. These calls are automatically generated internally by VISION:Builder for retrieval and updating.

Calls are made to DB2 by VISION:Builder as rows are retrieved to build a logical record. See Chapter 3, *Defining Data* for details about how a logical record is built and its relationship to relational databases. During the build process, VISION:Builder issues an OPEN CURSOR statement (for each segment in the logical record) and issues FETCH statements as appropriate to retrieve a row (segment occurrence). The cursor is governed by a SELECT statement that is created by VISION:Builder. The SELECT statement identifies the fields that are to be retrieved from the relational database together with any qualification or restriction against the rows of the table that are to be retrieved. These qualifications and restrictions are defined to VISION:Builder by the logical relation in the file definition. In practice, the logical relations are turned into WHERE clauses in the SELECT statement.

Sometimes it is necessary to provide constraints or WHERE clauses that cannot be defined in the logical relation. These provisions are made through the “sql-where-clause“ specified on the FILE command SEGMENT and WHERE keywords . This provides you the ability to extend the WHERE clause as created by VISION:Builder by connecting any appropriate extension that you choose to provide. WHERE clause extensions can be provided for any segment in the logical record; they are not limited to the root segment.

The cursor remains open until all occurrences of that segment type have been built in the logical record. At that point, VISION:Builder issues a CLOSE CURSOR statement.

VISION:Builder issues calls to DB2 to write a row to a relational table. Each row corresponds to a particular segment occurrence. This requires either an INSERT or UPDATE statement. Rows can also be deleted by VISION:Builder, requiring a DELETE statement. See Chapter 15, *Advanced Transaction Processing* for information on updating a relational files.

The following discussion shows how VISION:Builder constructs these DB2 statements.

### SELECT Statement

When the SQL keyword is not present in a FILE MASTER or FILE CORDn command, VISION:Builder generates the required DB2 SELECT statement. When the SQL keyword is present in these commands, the SQL keyword text is used for the DB2 SELECT statement.

The general format of a SELECT statement generated by VISION:Builder to retrieve a row is:

```
SELECT set of fields FROM table_name X1
WHERE logical relation   AND
                           WHERE keyword text

SELECT set of fields FROM table_name X1
WHERE logical relation   AND
                           WHERE keyword text
ORDER BY key fields
```

The general format of a SELECT statement generated by VISION:Builder to retrieve a row that matches the segment key values is:

```
SELECT set of fields FROM table_name X1
WHERE key1 = ? AND ... keyn = ?  AND
                    logical relation       AND
                          WHERE keyword text
```

The set of fields consists of all key fields in the segment, plus all other fields referenced in the application (for example, in a procedure, report, or transaction definition). Any fields in the file definition that do not fall into this category are omitted from the SELECT statement and, therefore, are never retrieved from DB2. This adds considerably to the efficiency of your application.

If the segment is defined with ascending or descending order, an ORDER BY clause containing key fields is appended after the WHERE clause.

The X1 refers to the correlation name for the table. This is sometimes necessary to distinguish identically named columns that come from different tables.

For a direct-retrieval by key, an expression for each key field (key1 through keyn, where n is the number of keys in the segment) is included in the WHERE clause.

If a segment is a multi-column keyed segment, there are a few exceptions to the above guidelines. See Chapter 3, *Defining Data* for information about defining a multi-column keyed segment. Only one key field can be defined in a multi-column keyed segment and it does not have a DB2 name associated with it. VISION:Builder, therefore, must omit it from the SELECT statement. The ORDER BY clause is omitted as well and the user must supply it, if needed, on a "sql-where-clause" following the WHERE keyword on the FILE command.

### SELECT Statement with Column Functions

The general format of a SELECT statement generated by VISION:Builder to retrieve a row for a segment that contains column functions is:

```
SELECT set of fields, column function(s) FROM table_name X1
WHERE logical relation   AND
                          WHERE keyword text
GROUP BY set of fields
```

The set of fields consists of all fields defined in the previous SELECT that are *not* column functions. This set of fields is repeated on the GROUP BY clause, specifying the fields for which you want summarized values. Because the set of fields includes all key fields, picking a unique key field for the segment will nullify the effect of the GROUP BY. Therefore, the key field chosen for the segment should not uniquely identify the segment. See Chapter 3, *Defining Data* for an example of defining column functions in a relational file definition.

### INSERT Statement

The general format of an INSERT statement generated by VISION:Builder to insert a row is:

```
INSERT INTO table_name
(set of fields) VALUES (?,?,...?)
```

The set of fields consists of *all* fields defined in the file definition. If any columns in the table are defined as NOT NULL, they *must* be defined in the file definition in order for the INSERT to be successful. If defined as NULL or NOT NULL WITH DEFAULT, they can be omitted from the file definition.

### UPDATE Statement❹

The general format of an UPDATE statement generated by VISION:Builder to update a row is:

```
UPDATE table_name X1
SET set of fields = ?
WHERE keys = ?      AND
                              logical relation
```

The set of fields consists of all non-key fields that are referenced in the run. Any fields in the file definition that do not fall into this category are omitted from the UPDATE statement and, therefore, are not updated in the table.

The keys consist of all keys (up to nine maximum) that are defined for the segment. These, along with the logical relation, help to prevent a set of rows from being updated. However, if the keys you choose do not uniquely identify a row, an entire set of rows is updated by DB2.

### DELETE Statement❹

The general format of a DELETE statement generated by VISION:Builder to delete a row is:

```
DELETE FROM table_name X1
WHERE keys = ?      AND
                              logical relation
```

The keys consist of all keys (up to nine maximum) that are defined for the segment. These, along with the logical relation, help to prevent a set of rows from being deleted. However, if the keys you choose do not uniquely identify a row, an entire set of rows is deleted by DB2.

## Uniqueness of Rows in Relational Tables

Normal operation against relational tables will permit duplicate rows to be entered into the table (for example, by the INSERT statement). You can prevent duplicate rows by using the DB2 UNIQUE INDEX clause to create a unique index for each inserted row.

## Foreign Keys

When you are performing an insert segment action on lower level segments through transaction processing,❹ define the foreign key (from the logical relation) as a field in the file definition. Remember that you are not required to define fields used in the logical relation in the file definition. If this field is not provided with data from the transaction, it is necessary to provide the value of this field in a request. The intent is to have the foreign key column and its contents included in the row when the row is written because the foreign key is the way DB2 relates the rows of this table to another table.

## Validation Against the DB2 Catalog

No validation against the DB2 catalog takes place during the VISION:Builder file definition run. When cataloging a file definition, DB2 does not need to be available. This also means that any undefined relational table and column names are not detected.

When processing an application, VISION:Builder is connected to DB2 during the compile and process phases of the run. This enables any errors in the file definition or the generated DB2 statements to be detected during the compile phase. A scan run, therefore, is helpful in correcting these errors in your application.

## COMMITs During VISION:Builder Processing

There are three points during processing when VISION:Builder automatically calls DB2 to perform a commit operation.

- When replacing a subfile, if TABLESPACE is not specified on the EXTRACT TABLE command, a commit operation is performed between the drop operation and the create operation.
- When creating a subfile, VISION:Builder performs a commit operation after the drop operation and create operation of the relational table.
- At the end of a job that completed successfully.

If commits are wanted at more frequent intervals, you must specify the intervals on a CHECKPOINT command in the Run Control command group. At the time a checkpoint is taken, VISION:Builder also calls DB2 to perform a commit operation. If you specified commit only without checkpoints on the Checkpoint statement, only the COMMIT statement is executed at the specified interval.

## Performance Considerations

The bulk of the resources in a VISION:Builder application involving relational tables are consumed by the relational database manager, DB2. It is important to know that VISION:Builder utilizes dynamic DB2 statements and generates them based upon the specifications provided in the file definition and the fields used in the application. So you do not want to ask VISION:Builder to generate DB2 statements that may have adverse performance effects. For example, the Suppress Duplicates and the Segment Order specifications for a segment in the file definition are merely a Yes or No entry; yet, they cause VISION:Builder to generate DISTINCT or ORDER BY clauses that require DB2 to perform extra processing and sort the selected rows.

VISION:Builder does not automatically perform expensive DB2 joins. It issues tailored SELECT statements. The correct combination of SELECT statements within a logical record has the same effect as if an outer join were performed. The logical record makes the capability of defining outer joins a trivial process.

VISION:Builder is optimized to map key fields and only those fields that are actually used in the application, providing for greater efficiency.

The definition of indexes is external to VISION:Builder. DB2 always honors a request for data whether or not an index exists. If the index does not exist, an entire database scan is performed. This can be very costly. You are advised to review whether foreign keys (that is, fields referenced on the left hand side of a logical relation expression) should be indexed. Fields defined as VISION:Builder key fields are also candidates for an index. These decisions must be made in the light of the application and the nature of the tables involved. Too many indexes negatively impact the application as they are updated.

The EXPLAIN function can be useful in determining the efficiency of an application that uses relational databases. It causes DB2 to write performance information out to a plan table that you can later list. This function is specified as a keyword on the CONTROL command and included in the run control group. See the *VISION:Builder ASL Reference Guide* for details on the EXPLAIN function.

## NULL Columns

The intent in VISION:Builder is to preserve the NULL status of a column when retrieving and updating❹ a row and to create the NULL status of a column when inserting a row.

This applies to any field in a MOSAIC (memory optimized storage) relational coordinated file or master file or relational subfile that is defined as NULL to the relational database manager. VISION:Builder flags it as an invalid field (that is, it prints as an asterisk (*) or other installation-specified character) upon retrieval.

Throughout transaction❹ and procedural logic processing, the invalid flag is only reset in response to a change in the contents of the field. At output time (this applies only to the master file and subfiles), an invalid field is written to the table with a NULL status. In this fashion, a column that was NULL when retrieved from the table is written back out to the table as NULL if the application did not update the field. When inserting a row, you can effectively set a column to NULL in the table by replacing an invalid value into the VISION:Builder field. When reporting fields from a MOSAIC relational file, a NULL column prints as an asterisk (*). This is different from NULL fields in a relational file that are not MOSAIC. Whenever a NULL field is mapped into the logical record, it is converted to blank or zero, as appropriate.

## Multi-Column Keyed Segments

Chapter 3, *Defining Data* discusses how to define multi-column keyed segments in the file definition; this section discusses the circumstances under which an application can access these segments.

A multi-column keyed segment is defined in a relational file definition when it requires more than one column to uniquely identify a row. When a segment is defined in this manner, you can access these segments based on multiple column values in instances where VISION:Builder allows only one key.

When VISION:Builder performs a direct-read of any segment in a file, it is restricted to only one key, the primary key. As a result, the row can only be accessed by one column. With a multi-column keyed segment, you define a set of columns whose values are used to retrieve a particular row, while still retaining one key in the definition.

Relational file definitions containing multi-column keyed segments can be used in a VISION:Builder application (in conjunction with supplemental WH statements that you provide) for processing coordinated files that require direct-read access of any of their segments. Direct-reads with multi-column segments are allowed in the following instances:

- When reading the root segment of an indexed coordinated file.
- When reading the root segment of a user coordinated file processed with the FIND function in ASL or the RE/RG operator in fixed format syntax.
- When reading any lower level segment of a MOSAIC coordinated file that is accessed with the qualified FIND function in ASL or a qualified Find Segment (FS) operator in fixed format syntax (that is, a FS operator that specifies a key value in operand B).

Multi-column keyed segments are supported for relational coordinated files that are accessed either in standard processing mode or MOSAIC processing mode. They must still meet the above criteria.

These types of segments are not supported for relational master files.

The host variables on the supplemental WHERE keyword operands defined using an "sql-where-clause" on a FILE command do not start a loop, even though they are VISION:Builder fields. VISION:Builder assumes that any host variable has already been referenced in the application and, therefore, contains the required data value. If a host variable resides in a lower level segment, make sure that your application has started a loop on that segment prior to any access of the multi-column keyed segment.

When using the O, N, and blank qualifiers for the master file, make sure you are consistent. Separate loops are started on the old and new master file segments. If the match field for an indexed coordinated file uses a blank or N qualifier, make sure that the WHERE "sql-where-clause" for that segment uses the same qualifier for the host variables.

VISION:Builder cannot generate a correct ORDER BY clause for the key of a multi-column keyed segment because there is no corresponding column name for the key. You are required to code the necessary ORDER BY clause on a WHERE "sql-where-clause" in the following instances:

- When processing a user coordinated file with the qualified FIND function followed by an unqualified FIND function. In fixed format syntax, when using the RE or RG operator followed by subsequent RD operators.
- When reading a lower level segment by key value using a qualified FIND function followed by a subsequent unqualified FIND function. In fixed format syntax, when using a qualified Find Segment (FS) operator followed by subsequent unqualified FS operators.

When using qualified FIND functions without MOSAIC processing, VISION:Builder does not perform a direct-read of the segment, but multi-column keyed segments can still be effective. All segments reside in memory so

VISION:Builder merely searches the segments for the key value. However, because the key has been overdefined by component parts that map to multiple columns, a multi-column keyed segment searched in this manner finds segments based on multiple column values.

Chapter

# 17 Report Manager

The report manager facility provides a mechanism that allows you to control the ordering, grouping, and destination of entire reports or subsets of reports produced by VISION:Builder applications. The report manager facility consists of two major functions referred to as collating and routing. The functions are specified using the ASL commands COLLATE and ROUTE in the Run Control command group.

The report manager collating facility allows you to enter specifications to subdivide and regroup the report information into collation sets by identifying entire reports by their name, or subsets of reports by name and sort field information. This collation group information, reports or subsets of reports, is then output in the specific order designated by the COLLATE commands.

The collation process packages each report or report subset as a separate report, complete with its own preface page, individual page numbers beginning with 1, and totals computed from the information for that subset alone.

The report manager routing facility allows you to enter specifications to control the output destination of entire reports or subsets of reports produced by VISION:Builder applications. The reports and/or report subsets can be directed to separate and/or multiple output destinations. Entire reports are identified by name and subsets of reports are based on the report sort field information.

The routing process packages each report or report subset as a separate report, complete with its own preface page, individual page numbers beginning with 1, and totals computed from the information for that subset alone.

The routing capability allows you to tap the full facilities of your operating system and its communication network to direct reports and report subsets to remote sites, microfiche devices, special forms, multiple copies, tape, disk, and so on.

The report manager facility collating and routing capabilities provide you with a variety of ways to deliver the report output information.

- Use COLLATE to control the order in which the reports are output.
- Use COLLATE to create subsets of information from reports.
- Use ROUTE to direct a single report to one or more destinations.
- Use ROUTE to direct subsets from a single report to one or more destinations.
- Use ROUTE to direct any number of reports and/or subsets to any number of different destinations.
- Use ROUTE to direct any number of reports and/or subsets to the same destination.
- Combine COLLATE and ROUTE to reorder and/or create subsets of report outputs and then direct the restructured report output stream to any variety of destinations.
- Use COLLATE and/or ROUTE to package each report and/or subset as a separate report, complete with its own preface page, individual page numbers beginning with 1, and totals computed from the information for that group alone.

  **Note:** Only the reports identified in the COLLATE and/or ROUTE commands are handled by the report manager facility. Any report not handled by the report manager is directed to the default destination identified by the M4PRINT JCL statement.

  If you include a JCL statement named (DD or DTF name) M4PRINT without any other report manager commands, all the application report output is automatically directed to the M4PRINT statement for output. Alternate list report output is directed to its designated destination. The source listing output is directed to M4LIST as always.

- Use your operating environment parameters on the JCL statements to exploit all the facilities within your network for delivering and distributing report output to remote printers, remote locations, special forms queues, microfiche, tapes, data sets, and so on.
- Monitor reporting activity to remote locations by reviewing the summary page on M4LIST that shows the number of pages and lines that were routed to each destination.

The VISION:Builder report manager facility operates in both the single-step and three-step application runs. There are no additional job steps required. Except for an increase in virtual storage, essentially no additional computer resources are required.

The report manager is not sensitive to the report format. You do not need to make any changes to the application procedures or requests that generate the reports to use the report manager capabilities.

# Reporting Concepts

The reporting elements that relate to the report manager facility and its process are identified and explained here for the purposes of collating and routing application reports and information. For more information on reporting, see Chapter 9, *Reporting* and Chapter 14, *Extended Reporting Capabilities*.

## Reports

Reports contain information developed within an application run from the specified input files, tables, arrays, and processing procedure and request commands and statements. Each report is identified by the name of the procedure/request within the application that contains the report specifications. Additionally, when fixed format statements are used to specify application reports, the n number from the Rn statement identifies a report within the named procedure/request. All references to reports in ASL commands are by name. When the keyword ALL is specified, it is used to identify all the reports in the application as a single group. All the information and data needed to generate each formatted report is directed to a report file for sorting, then reporting.

## Report Files and Sort Keys

A report file contains all the information and data needed to produce each formatted report specified in an application. The report file must be sorted into sequence prior to generating the actual formatted reports. A sort key is used to order the information in the report file. The first part of the sort key is a report number, which is an internal number assigned to each report based on its relative position within the application. The remainder of the sort key consists of record type identifiers and the actual data values from the fields designated as the sequence fields for the report and is referred to as the report key.

## Report Key

The report key consists of the field or set of fields designated as the sort sequence fields for a report. To form the report key, the fields specified on the ORDER command for a report are concatenated, based on their major to minor (1 to 9) sort sequence specification. The records on the report file that contain the designated report data values are sorted into report key order within each report.

## Collating Key

When report manager performs collation, a collating key is established. The collating key can be the entire report key or a subset, starting from the first position of the report key and extending for a specified number of positions as indicated by the COLLATE command KEYLENGTH parameter. The collating key is viewed as

a number of positions (bytes) of the report key from zero and extending up to and including the last position of the report key. Collation occurs based each change (control break) within the specified collating key length. When the collating key length is zero, no changes (control breaks) within the collating key will occur.

## Routing Key

When report manager performs routing, a Routing Key is established. The Routing Key is a subset of the report key starting from the first position and extending for a defined number of positions based on the ROUTE command KEYVALUE parameter specifications. The Routing Key is viewed as a number of positions (bytes) of the report key from zero and extending up to and including the last position of the report key. Routing occurs based on the presence of specific data values within the routing key length as specified in the ROUTE command KEYVALUE parameter. When the routing key length is zero, routing occurs for all data values (the entire report).

Figure 17-1 illustrates the relationships of the elements within the report file that are used by report manager to perform collating and routing.

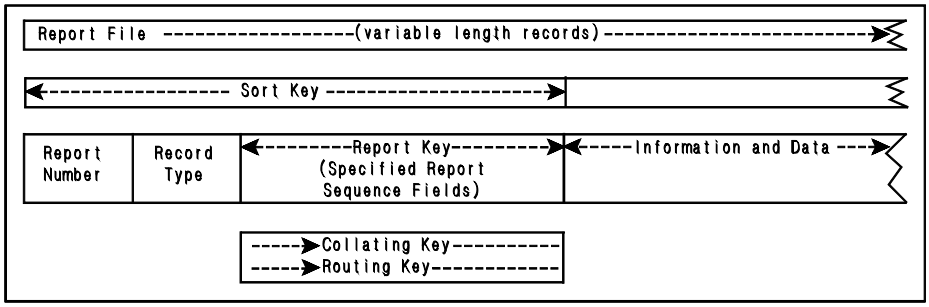



Figure 17-1 Report File Elements

Note that the relationship of the report file elements and where the collating and routing keys are used in the identification process for collating and routing. The report file is ordered by the sort key. Each individual report set has its information and data ordered further by its report key. The collating and routing key specifications are applied to the ordered report key.

## Reporting and Report Manager Sequence of Events

During application processing and reporting runs, there is a sequence of events that occur in a specific order. The events that are related to reporting and the Report Manger facility occur in the following order.

- In the decode and compile portion of the application run, the report generating information for each report in the application is directed to the report file.
- In the processing portion of the application run, the actual data to be reported by the application is directed to the report file.

- At the end of processing, when all the input and the procedure/request processing has completed, the sorting occurs.
- During the input processing of the sort, the report numbers are adjusted to reflect the order and groupings specified by any COLLATE commands.
- During the output processing of the sort, the ordered report numbers and data are subdivided further based on changes (control breaks) in any specified collating keys.
- In the report generating portion of the application run, the reports or subsets are directed to the destinations based on any ROUTE command specifications.

> **Note:** The ROUTE command specifications are applied against the report file after it has been sorted, ordered, and subdivided, based on the application report sort sequence and COLLATE command specifications.

## Report Collating Overview

The concept of collating reports is not new. Once the reports are produced by an application, they are often manually subdivided and regrouped into packages that are delivered to the persons or departments requesting the information. This repackaging is done by report names and even by information within a report. The report manager facility automates this process of collecting the associated subsets from the application reports and regrouping these subsets into the specified packages.

Figure 17-2 illustrates how the standard report outputs are reordered and subdivided using the following simple COLLATE command:

```
COLLATE REPORTS Report2 Report1 KEYLENGTH 1
```

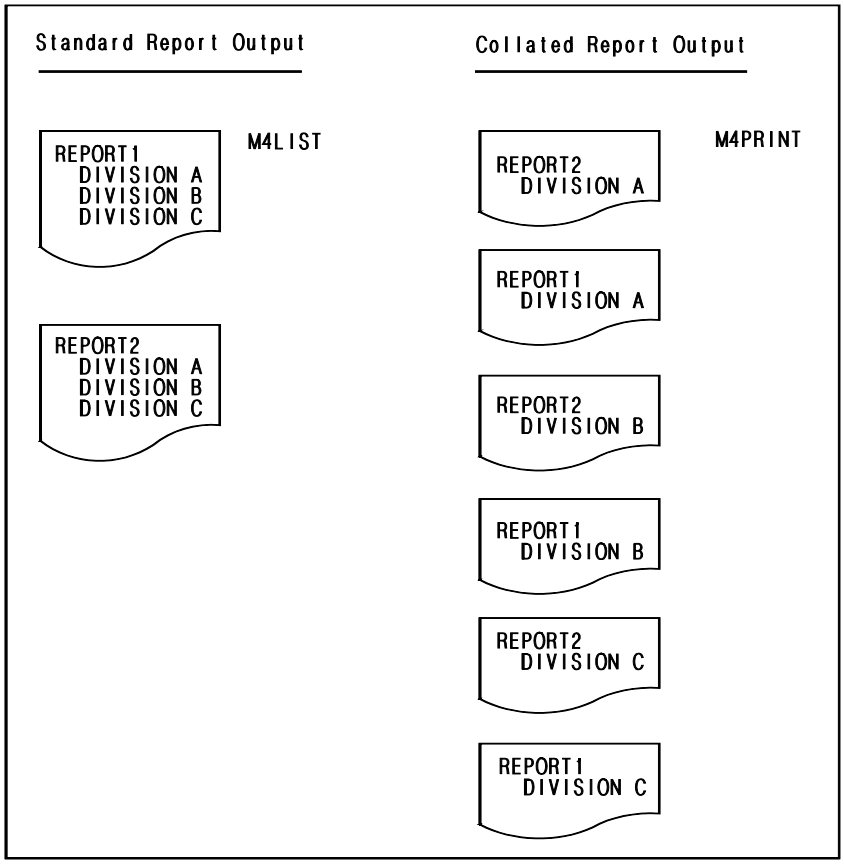



Figure 17-2 Report Collating

In Figure 17-2, two reports are produced by an application and are in sequence by the division field information within the reports.

The left side of Figure 17-2 shows the standard report output produced by VISION:Builder and directed to the JCL statement identified as M4LIST for output disposition. Report1 is followed by Report2, each containing the reported information for all the divisions in sequence by the division field.

On the right side of the Figure 17-2 is the collated set of the same report output. The COLLATE command specifies that Report2 should be output first followed by Report1, and the reports in the collation group should be subdivided by changes in the first position of the report key data values (that is, division).

Report manager automatically reorders the reports and prepares a separate report for each change in the division value. In the example, six separate reports are produced in the collated output, one for each unique division field value (A, B, C) within each named report.

The collating mechanism establishes each report subset as a separate report, complete with its own preface page, individual page numbers beginning with 1, and totals computed from the information within that subset alone. When only collating is performed, all reports are directed to the JCL statement identified as M4PRINT.

## Report Routing Overview

The concept of routing reports is the distribution of the output. Once the application reports have been produced, they are subdivided and regrouped into packages that are distributed to the specific location of the persons or departments requesting the information. This repackaging is done by report names and even by information within the reports. The report manager facility automates this process of gathering the associated subsets from the application reports and then directing these packages to their destinations, which are specifically named JCL statements. The actual disposition of the directed output is controlled by your operating system and the JCL statement parameters. With the report manager facility, you can route the application reports and report subsets to separate and multiple output destinations (JCL statements). Entire reports, as well as subsets, can be routed based on specific report information values.

Figure 17-3 illustrates how the standard report outputs are subdivided and directed to specific destinations by the use of the following ROUTE commands.

```
ROUTE REPORTS  Report1 TO Atlanta Office31
ROUTE REPORTS  Report2 KEYVALUE 'A' TO Dallas Office31
ROUTE REPORTS  Report2 KEYVALUE 'B' TO Atlanta
ROUTE REPORTS  Report2 KEYVALUE 'C' TO Office31
```

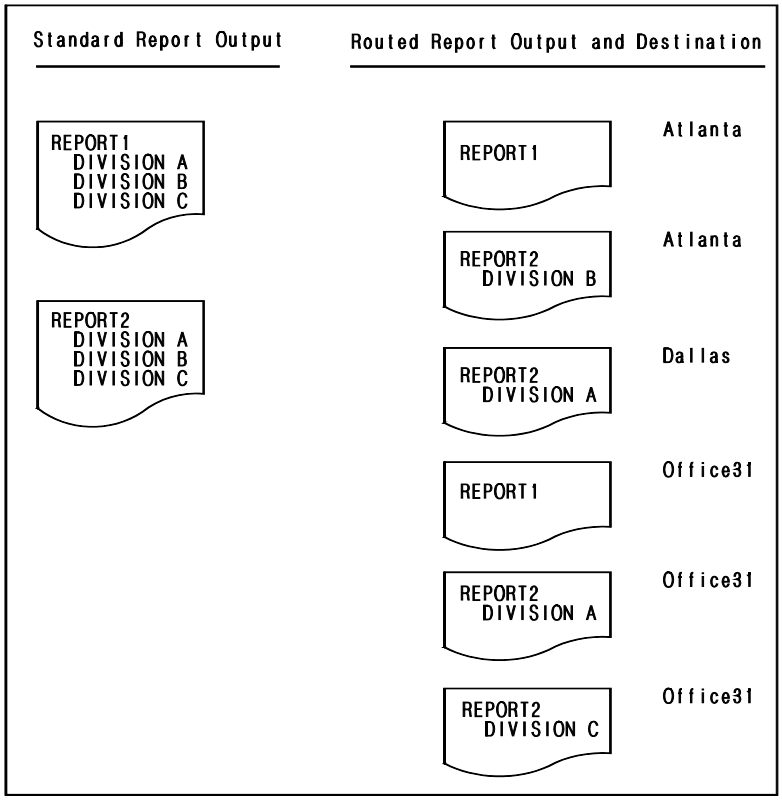



Figure 17-3 Report Routing

In Figure 17-3, two reports are produced by an application and are in sequence by the division field information within the reports.

The left side of Figure 17-3 shows the standard report output produced by VISION:Builder. Report1 is followed by Report2, each containing the reported information for all the divisions in sequence by the division fields.

On the right side of the Figure 17-3 is the routed set of the same report output. The ROUTE commands specify the destinations where the reports or subsets are to be directed. (Note that the destinations refer to JCL statements that contain the appropriate output directing parameters for your operating environment.) An entire copy of Report1 is routed to Atlanta and Office31. The Division A information from Report2 is routed to both Dallas and Office31. The Division B information from Report2 is routed to Atlanta. The Division C information from Report2 is routed to Office31.

Report manager automatically directs the designated reports or subsets to the specified output destinations. In the example, six separate reports/subsets are directed to three different destinations. Note that any reports not referenced for routing are automatically directed to the JCL statement identified as M4PRINT.

The routing mechanism establishes each report/subset as a separate report, complete with its own preface page, individual page numbers beginning with 1, and totals computed from the information within that report/subset alone.

# Collating with Routing Overview

Figure 17-4 illustrates the interaction of combining the following collating and routing functions:

```
COLLATE REPORTS Report2 Report1 KEYLENGTH 1
ROUTE REPORTS  Report1 TO Atlanta Office31
ROUTE REPORTS  Report2 KEYVALUE 'A' TO Dallas Office31
ROUTE REPORTS  Report2 KEYVALUE 'B' TO Atlanta
ROUTE REPORTS  Report2 KEYVALUE 'C' TO Office31
```

The illustrations from Figure 17-2 and Figure 17-3 have been combined.

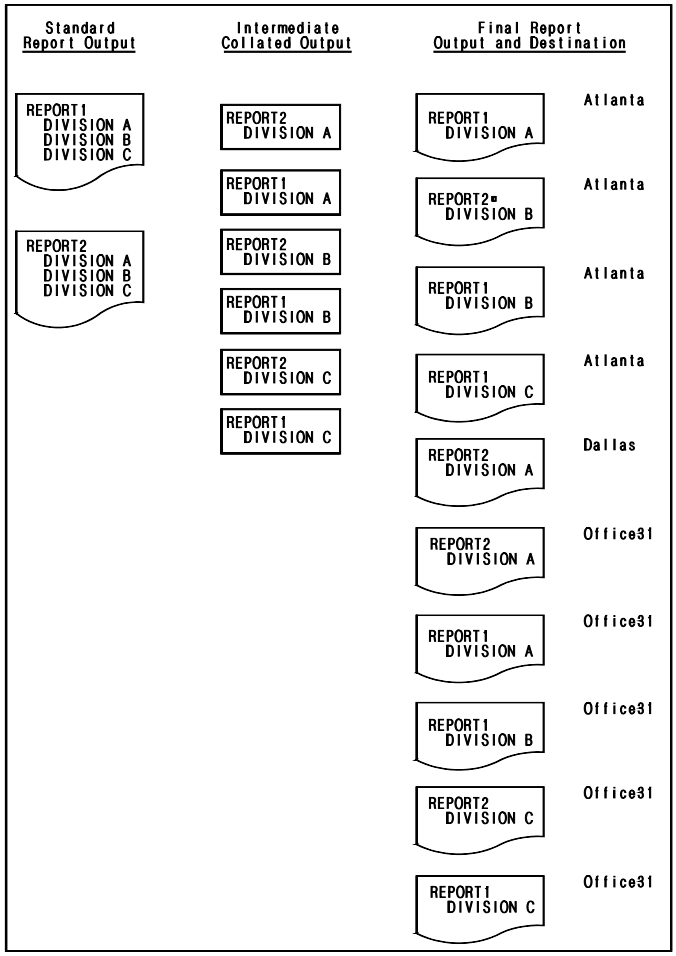



Figure 17-4 Report Collating and Routing Combined

Note that the collating process occurs before routing process. In Figure 17-4, there is an intermediate stage where the report output from the COLLATE command exists. This looks just like the collated output shown in Figure 17-2. At this point, the sort output is complete.

The ROUTE commands are applied to this intermediate stream of output reports. It is clear that the routed report output is different then what was shown in Figure 17-3. The reason for the difference is that Report1 is no longer a single report containing all the divisions A, B, and C. This is because the COLLATE command has subdivided Report1 and Report2 in separate reports for each division. When the entire Report1 is routed, each of the subdivisions of Report1 is routed.

In this illustration, the ROUTE commands are being applied to the six reports created by the COLLATE command. So when the named report, or named report and key value are encountered in the report output stream (the intermediate collated output), the report gets routed to the designated destination.

## Report Manager Specifications

The report manager functions are specified using the ASL commands COLLATE and ROUTE in the Run Control command group.

When using a three-step application run (SORT EXTERNAL), the report manager statements are input with the application source code in the process step. The COLLATE commands are passed to the sort step (actually, you execute the COLLATOR program, which calls the sort) along with the sort control statements in the data set pointed to by the M4SORT JCL statement. The ROUTE commands are placed on the report file (M4REPO and/or M4REP2 through M4REP9) for use in the report step. See the *VISION:Builder for OS/390 Environment Guide* or the *VISION:Builder for VSE Environment Guide* for more information about execution JCL.

For more information about coding ASL procedures and the report manager commands, see the *VISION:Builder ASL Reference Guide*.

## Considerations of Collating and Routing

The following topics discuss special considerations that must be taken into account when the report manager is activated.

Activation of the report manager may have an impact on all reports even though a specific report was not referenced on any COLLATE or ROUTE command. The following operating characteristics of the report manager should be kept in mind:

- When ALL is used to COLLATE or ROUTE every report in an application, every report is considered as being selected (that is, no reports perform default actions).
- Each COLLATE command establishes a set of reports, ordered as specified. The order of the COLLATE commands establishes the order in which the report sets are output. Therefore, the report output order is controlled by the COLLATE command order and the order of the reports in each command.
- When collating is specified and there is only one report in the application, the collating specification is effectively ignored. In other words, a collating specification cannot be used to split a single report into multiple reports.
- Any report not referenced on any COLLATE command is left as is with no collating (control) breaks. They will be the last reports in the M4PRINT data set.
- Any report not referenced on any ROUTE command is left as is and will be directed to the M4PRINT data set.
- When collating is specified for the purpose of reordering the reports (no KEYLENGTH specified), each report stays as is, only reordered.

  However, when collating includes a KEYLENGTH keyword, each entire report can be subdivided into many reports. A separate report occurs each time the report names changes. The report names change whenever the report key values change within the KEYLENGTH keyword and more than one of the reports within the collating group has the same report key values. Note that changes in the report key values within the same report do not cause separate reports to be generated. When collating reports together in the same group and using the KEYLENGTH keyword, each report should contain the same report key values for the portion described by the length parameter value.
- When routing is performed on reports and/or report subsets, a separate report occurs for each report and/or report subset directed to the same destination.
- When a separate report is produced by VISION:Builder, each report contains a complete set of elements associated with a complete and separate report. Depending on the various reporting options used, each separate report will have its own preface page, page numbering starting with 1, and summaries based only on the information in the report or subset. Each multiple image report will always start on a new physical page.
- When a report is referenced on a COLLATE or ROUTE command, you cannot specify alternate list for that report. Report manager specifications and alternate list are mutually exclusive.

- The presence of a M4PRINT JCL statement *without any other* report manager specifications in the application causes all report output normally directed to the M4LIST JCL statement to be directed to M4PRINT. Note that if alternate list is being used, report output messages go to M4PRINT and the reports go to the designated alternate list JCL statement.
- When the DEFER keyword is specified on a ROUTE command, the data set open is deferred until just before the first output is written to the destination. Depending on the operating environment, the JCL specifications, device types, and data set characteristics, different actions occur when no output is written to the destination. For tape data sets, DEFER prevents an empty data set by avoiding the tape mount and the open. For PDS members, DEFER prevents the creation of an empty member. For disk data sets, DEFER avoids the open and output; however, the operating system and access method mechanisms can still perform allocations or some handling based on the JCL specifications for the data set.

## JCL Considerations

- You must include a DD or DTF JCL statement for the default routing destination, M4PRINT. This is used when your application contains any report manager (either collating or routing) specifications. If the M4PRINT JCL statement is missing, report output for M4PRINT is directed to the M4LIST JCL statement instead.
- If your application contains report routing specifications, then you must also include a DD or DTF JCL statement for each routing destination mentioned in the application.
- If the application is run in the single-step mode, the destination DD/DTF statements are included in the job step that executes the application (because there is only one step).
- If the application is run in the three-step mode, the destination DD/DTF statements are included in the report step (the third step of the process/sort/report JCL).
- When collating specifications are contained in an application that is run in the three-step mode, change the name of the sort program executed in the sort step (the second step of the process/sort/report JCL) to COLLATOR. Note that the COLLATE commands are passed to the sort step (COLLATOR) as part of the SYSIN data set (M4SORT output from the processing step).

## Multiple Report Files

With three-step processing, the report step can process multiple input report files in one execution. Each report file must be sorted before being input to the report step. To accomplish this, concatenate the sorted report files in the wanted order at the M4REPI JCL statement.

The report files come from single or multiple processing steps. Each standard report file (M4REPO) and alternate report file (M4REP2 through M4REP9) can contain their own set of ROUTE commands passed from the processing step. The report step resets the routing criteria based on each set of ROUTE commands encountered on each report file. If a report file does not contain any routing commands, standard reporting is applied to that set of reports. Information can be combined and distributed from several systems or subsystems in a single report distribution process.

Note that when concatenating report files into a report step, you must supply the M4PRINT JCL statement to activate the report manager mechanisms, especially if the first report file does not contain routing specifications.

Another important use of this feature is reduce the computer resources used for sorting report files. Assume, for example, there are four large reports generated in a VISION:Builder process-sort-report run. Direct each report to a separate report file in the processing job step (M4REPO, M4REP2, M4REP3, and M4REP4), sort the report files individually, and bring the four report files into a single report phase job step using report manager. The total computer resources required to sort all four report files individually can be significantly less than attempting to sort the data for all reports in a single sort.

### Storage Considerations

When the report manager is activated, your region size for storage below the 16MB line may need to be increased by approximately 20K to accommodate the additional storage required by the report manager. The report manager also uses storage above the 16MB line. The amount of storage used above the line depends upon the complexity and number of reports. The average amount is in the 200K to 300K range.

## COLLATE and ROUTE Application Examples

Appendix D, *Complete Sample Applications* contains listings of application examples that demonstrate the use of the COLLATE and ROUTE Commands of the Report Manager Facility. This appendix contains the following application sample runs:

- Standard Reports Application on page D-180
- COLLATE Reports Application on page D-186
- ROUTE Reports Application on page D-193
- COLLATE and ROUTE Reports Application on page D-202

The VISION:Builder System Tape contains the Job Stream used to produce the listings. The EXAMPLES Data Set contains the R17FIGS member so you can run the examples as needed.

Chapter

# 18 External Program Linkage

User-written programs or system control programs (for example, database managers) can pass control to (call) VISION:Builder and VISION:Builder may in turn call external user-written programs or system provided programs (for example, database managers, utility programs, and subroutine libraries).

This chapter discusses the parameter conventions assumed by VISION:Builder when it is invoked either by the operating system or another program, as well as the coding requirements when a CALL command in a VISION:Builder procedure is used to call an external program.

## Linking to VISION:Builder from an External Program

User-written programs and system control programs can invoke VISION:Builder as a subprogram. Any parameters being passed to VISION:Builder by the calling program must be defined to VISION:Builder using linkage section definitions. The following sections discuss the linkage section conventions and how the linkage section is defined.

## Linkage Section Conventions

VISION:Builder accepts input parameters from any program invoking VISION:Builder as a subprogram. The input parameters are defined to VISION:Builder in the linkage section. In this discussion, input parameters refer to data elements whose space is allocated by another program (including the operating system) and whose location is passed to VISION:Builder by that program when it invokes VISION:Builder. For example, if your run is an IMS job, the parameters would be a set of PCBs. The operating system PARM field could also be defined as all or part of the linkage section.

Each parameter passed to VISION:Builder must be defined in a separate, single level VISION:Builder file definition. Each field that you want to reference in a VISION:Builder procedure must be defined. Because several definitions can be

combined into one linkage section, field names must be unique across all linkage section definitions. Fields defined in the linkage section are referenced in VISION:Builder procedures using the qualifier V.

### Parameter List Construction

Associated with the parameter list passed to VISION:Builder is an address list that is assembled and compiled as part of the calling program. There is a one-to-one correspondence between the addresses and parameters: the first address is the address of the first parameter, the second is the address of the second parameter, and so on. The address list is composed of a set of fullword addresses with the sign bit set to one on the last address.

When calling VISION:Builder, you specify which entry point to VISION:Builder you want to use: MARKIV or MARKGSI. Because VISION:Builder is not a reusable program, the calling routine must ensure that a fresh copy of VISION:Builder is loaded each time it is called.

The MARKIV entry point is usually specified when you want to pass PARM or DTF/DD name list information to VISION:Builder. If MARKIV is used and a parameter list is specified, the contents of the first four addresses in the list are fixed as shown in the following table.

**Note:** The specification of 00 00 00 for Word 1 and Word 2 can be a zero address or the address of a halfword field containing hex zero.

The specification for Word 3 and Word 4 can be a zero address or the address of a field containing a zero address with the sign bit set to one (hex '80 00 00 00').

| Word | Byte 1 | Bytes 2-4 |
|---|---|---|
| 1 | '00' | Address of PARM Field (or 00 00 00) |
| 2 | '00' | Address of DTF/DD Name List (or 00 00 00) |
| 3 | '00' | 00 00 00 |
| 4 | '00' | 00 00 00 |

Because the preceding example assumes more addresses will follow, you must provide null entries for words three and four, which are reserved for VISION:Builder internal use. Regardless of the number of entries in the parameter list, the last address in the list must always have the sign bit set to one. You may, however, need just the PARM information. In this case, the sign bit is set to one on the first address. Similarly, if both the PARM field and the DTF/DD name list are required, the sign bit of the second word is set to one.

The following address list is used to build a linkage section consisting of the PARM field and three user data areas (LINKA, LINKB, and LINKC). These four fields are defined in separate, cataloged, single level file definitions. The correspondence

between the file definition and the entries in the parameter lists is defined on the Files Used window of an application (RF statements in fixed format syntax). (See Defining the Linkage Section on page 18-5 for more information about defining linkage sections.)

In this example, the PARM field is AREA 1, the LINKA area is AREA 5, the LINKB area is AREA 6, and so on. Because the DTF/DD name list is not needed, no LINKAGE statement is supplied for AREA 2. The important point for you to consider is that the file definition and the last two characters in the AREA n specification are associated with the position of the address in the parameter list.

Sample LINKAGE statements for the address lists shown in the following table and the table starting on page 18-3 are shown in Figure 18-2 and Figure 18-3, respectively.

**Note:** This specification can be a zero address or the address of a halfword field containing hex zero.

This specification can be a zero address or the address of a field containing a zero address with the sign bit set to one (hex '80 00 00 00').

| Word | Byte 1 | Bytes 2-4 |
|---|---|---|
| 1 | '00' | Address of standard PARM field |
| 2 | '00' | Address of DTF/DD name list (or 00 00 00) |
| 3 | '00' | 00 00 00 |
| 4 | '00' | 00 00 00 |
| 5 | '00' | Address of LINKA |
| 6 | '00' | Address of LINKB |
| 7 | '00' | Address of LINKC (last entry in the list) |

In most cases, you will want to pass private parameters to VISION:Builder. In this case, the entry point MARKGSI can be used. When VISION:Builder is entered at MARKGSI, the format of the parameter list is defined entirely by the user.

For example, if you want to build a linkage section consisting only of the data areas LINKA, LINKB, and LINKC, a call using the MARKGSI entry point would produce the address list shown in the following table. In this example, AREA 1 refers to LINKA, AREA 2 refers to LINKB, and so on.

| Word | Byte 1 | Bytes 2-4 |
|---|---|---|
| 1 | '00' | Address of LINKA |
| 2 | '00' | Address of LINKB |
| 3 | '80' | Address of LINKC (last entry in the list) |

### VSE Considerations

VSE users who want to call VISION:Builder from an application program can use one of the following methods:

- MARKGSI can be linked with the application program and called as a relocatable program module.
- MARKGSI can be loaded at execution time by the calling program.

Parameters can be passed to VISION:Builder using the same type of address list described in the previous section for the MARKGSI entry point. Due to the nature of VSE, the following considerations must also be observed.

- Programs using the MARKGSI entry must run in a virtual partition divided into two areas (see Figure 18-1) by the SIZE parameter of the EXEC statement. VISION:Builder uses the remaining area in the defined virtual partition. Storage that is not needed by VISION:Builder can be made available for calling program use through the Freemain Size entry on the Application Parameters window (FREESIZE parameter on the RP statement in fixed format syntax).
- The application program phase name cannot begin with the letters MARK.

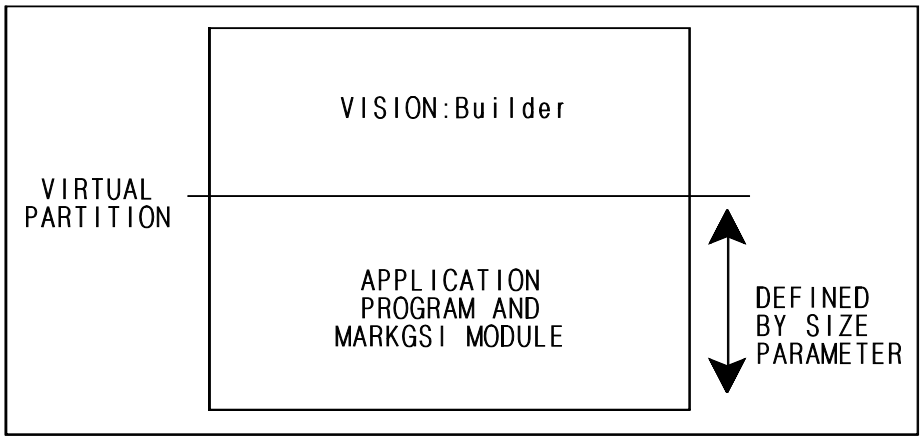



Figure 18-1 VSE Partition with MARKGSI

When using single-step sort, the interface routine MARKGSI should be replaced by MARKGSS.

# Defining the Linkage Section

Linkage sections are defined using the standard file definition mechanisms (see Chapter 3, *Defining Data*). Each parameter in the linkage section requires its own file definition.

In addition, these definitions must observe the following rules.

- The definition must describe an unstructured (single level) file.
- The definition cannot include variable length (type V) fields.
- When more than one definition is used to describe the linkage section (multiple parameters), all fields must be uniquely named across all definitions. Because all fields in the linkage section are qualified with V, the unique identification of a field must come from its defined name.

When using a linkage section, an ASL LINKAGE command in the Run Control command group or an RF statement in fixed format syntax must be used to relate the definitions to the VISION:Builder DTF/DD names of AREA n or M4LINKnn, respectively.

A V qualifier is used to access fields in the linkage section.

Figure 18-2 shows the LINKAGE commands necessary to build a linkage section that consists of the PARM field and three user data areas called LINKA, LINKB, and LINKC, and uses the MARKIV entry point. In this example, the PARM field is described by OSPARMFD and the data areas LINKA, LINKB, and LINKC are described by LINKAFD, LINKBFD, and LINKCFD, respectively.

```
CONTROL NAME LINKMIV
FILE MASTER INPUT  CUSTOMER
;
LINKAGE AREA 1, NAME OSPARMFD
LINKAGE AREA 5, NAME LINKAFD
LINKAGE AREA 6, NAME LINKBFD
LINKAGE AREA 7, NAME LINKCFD
```

Figure 18-2 Linkage Using the MARKIV Entry Point

If you specify the entry point MARKGSI, you could use the LINKAGE commands shown in Figure 18-3 to build a linkage section consisting of LINKA, LINKB, and LINKC (described by LINKAFD, LINKBFD, and LINKCFD, respectively). In both cases, there is a one-to-one correspondence between the LINKAGE AREA number and the position of the entry in the address list.

```
CONTROL NAME LINKGSI
FILE MASTER INPUT  CUSTOMER
;
LINKAGE AREA 1, NAME LINKAFD
LINKAGE AREA 2, NAME LINKBFD
LINKAGE AREA 3, NAME LINKCFD
```

Figure 18-3 Linkage Using the MARKGSI Entry Point

The LINKAGE commands in Figure 18-4 and Figure 18-5 show two ways in which a set of PCBs from an IMS file could be used to build a linkage section. Figure 18-4 shows the order as PCB1, PCB2, PCB3. Figure 18-5 shows the order as PCB2, PCB3, PCB1. Note that the parameter list passed to VISION:Builder must match the order specified by the LINKAGE AREA commands.

```
CONTROL NAME LINKDLI1
FILE MASTER INPUT  CUSTOMER
;
LINKAGE AREA 1, NAME LINKPCB1
LINKAGE AREA 2, NAME LINKPCB2
LINKAGE AREA 3, NAME LINKPCB3
```

Figure 18-4 Linkage Using the MARKDLI Entry Point, Example 1

```
CONTROL NAME LINKDLI2
FILE MASTER INPUT  CUSTOMER
;
LINKAGE AREA 1, NAME LINKPCB2
LINKAGE AREA 2, NAME LINKPCB3
LINKAGE AREA 3, NAME LINKPCB1
```

Figure 18-5 Linkage Using the MARKDLI Entry Point, Example 2

Figure 18-6 through Figure 18-9 show how the VISION:Builder file definitions might be coded using VISION:Workbench for DOS for these IMS data blocks.

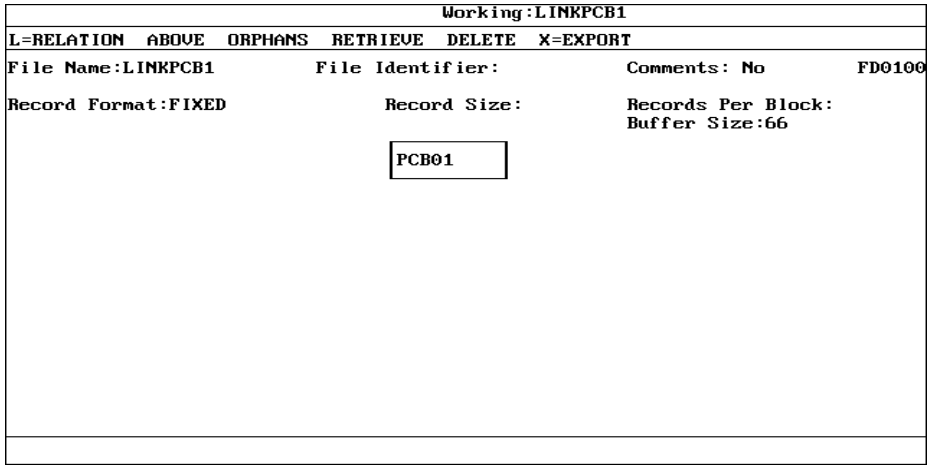



Figure 18-6 Sample Linkage Section, File Definition

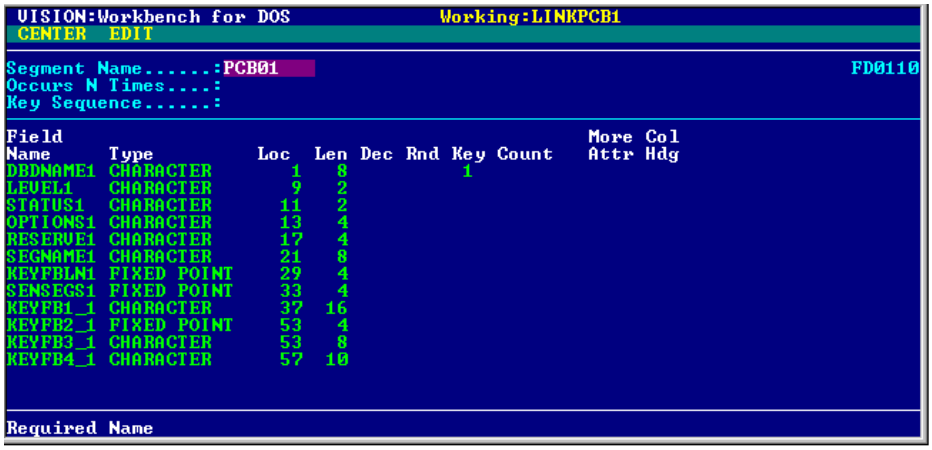



Figure 18-7 Sample Linkage Section, Segment Definition (PCB01)

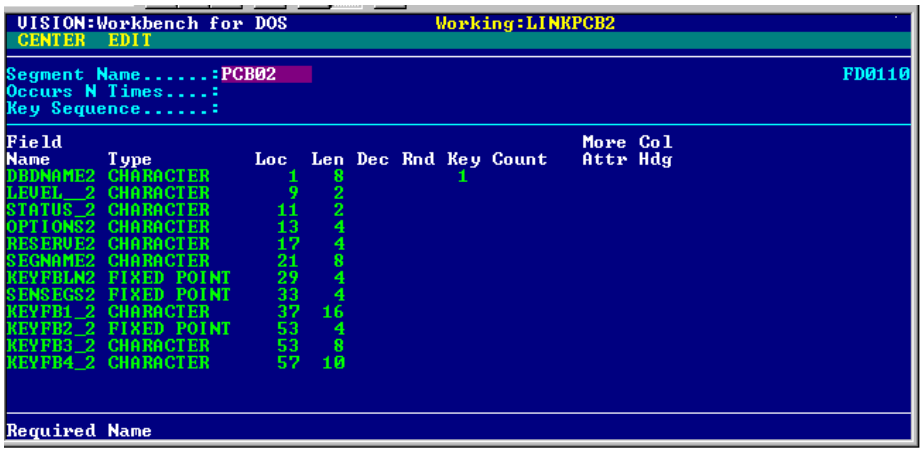



Figure 18-8 Sample Linkage Section, Segment Definition (PCB02)

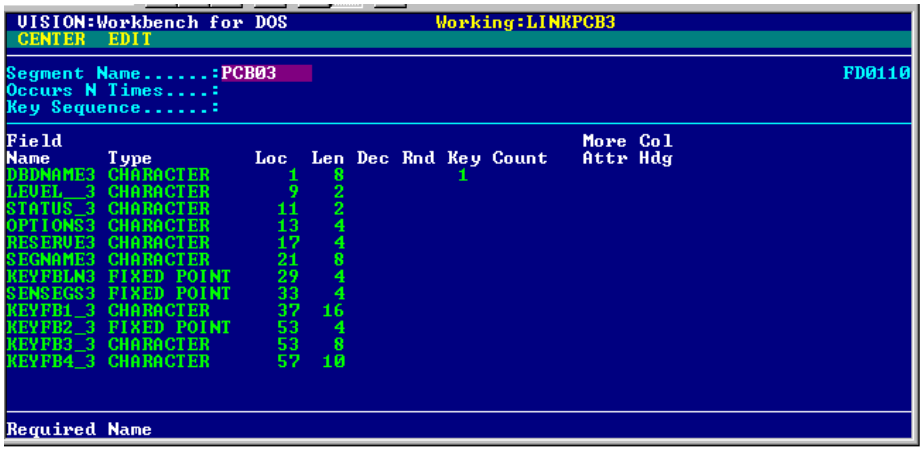



Figure 18-9 Sample Linkage Section, Segment Definition (PCB03)

# Calls from VISION:Builder to an External Program

External calls are specified using the CALL command in ASL (PR statement in fixed format syntax). External calls allow a VISION:Builder program to communicate directly with other systems and programs written in procedural languages. Data can be passed to and from VISION:Builder, once it has been defined to VISION:Builder using the procedure described in this section. See the *VISION:Builder ASL Reference Guide* or the *VISION:Builder Reference Summary* for more information about the CALL command.

Control can be passed from VISION:Builder to other systems or programs by means of CALL commands, which provide a flexible method of constructing a parameter list to be passed to a routine invoked by you. You have complete flexibility in specifying calls, which can be issued to any database system or user written routine.

Two additional flags are available for reference in VISION:Builder procedures to determine the status of a call and to provide call suppression information. Through the call, you can integrate the processing power of specialized stand-alone programs directly into VISION:Builder procedures, a capability otherwise available only through own-code. (A description of the own-code capability can be found in the *VISION:Builder for OS/390 Environment Guide* (OS/390 users) or the *VISION:Builder for VSE Environment Guide* (VSE users).) With the CALL command, you can generate the exact parameter list required by the external routine.

Routines invoked through the CALL command should access only those values passed as parameters and should not modify values or data structures of the VISION:Builder internal environment. In particular, fields specified as constants, key fields, and count fields should not be modified when passed as parameters.

Calls are well integrated into VISION:Builder, allowing, for example, the following to be specified as parameters: table lookup result fields, indexed coordinated file fields, fields from user-read coordinated files, transaction file fields,❹ time fields, and various flag fields. Calls can be made to database systems and can be specified in transaction❹ and subroutine procedures.

This capability requires a general knowledge of programming calls and interface techniques, as well as a familiarity with VISION:Builder.

## Specifying External Calls

In specifying a CALL command, you supply entries that form a parameter list for a routine to be invoked. In the calling list, you can specify as many parameters as necessary or just the name of the routine to be invoked. Calls are issued to database systems or user-written routines using a conventional parameter string format.

When a CALL command is encountered in a VISION:Builder procedure or request, it is assumed that all of the fields specified in the calling list have been previously initialized or established. The format of a CALL command in both ASL and fixed format syntax is illustrated in Figure 18-10.

```
EXTCALL: PROC
  CALL MODULE EPNAME USING 'CHARACTER STRING *$1' 'EMPLDATE'  ,
                           4  -899  9.700  'EMPLOYEE HISTORY'
END PROC

 ------------------------------------------------------------------------------------------------------
 STMT SEQ  LOG CON .....OPERAND-A.....  OPER  .............OPERAND-B.............     SEQUENCE   LIST
 TYPE NO.  LEV CTR QLF FIELD   SEG-LVL  ATN   QLF FIELD   SEG-LVL  (OR CONSTANT)     73 TO 80    SEQ
 ------------------------------------------------------------------------------------------------------
 (PR) (100)  (#)       (CALL EPNAME   C#CHARACTER STRING *$1 C#EMPLDATE D#4      ) (X)  (ASLGEN'D)     5
 (PR) (101)  (#)       (D#-899 D#9.700 C#EMPLOYEE HISTORY                        )      (ASLGEN'D)     6
```

Figure 18-10 Sample CALL in ASL and Generated Fixed Format Syntax

In fixed format syntax, a CALL statement is specified on a PR statement as follows:

| Position | Explanation |
|---|---|
| 1-8 | Must contain the procedure name. |
| 9-10 | Statement type – PR. |
| 11-13 | Optional sequence numbers. |
| 14 | The system delimiter (# in this example) entered in this position indicates that a call is being specified. The delimiter is repeated for each line of the call if it requires more than one PR statement. |
| 15-71 | Parameter string. Any number of parameters can be specified on as many PR statements as necessary. A parameter cannot be split between two lines. If a parameter does not fit in the remaining spaces of a statement (up to position [71]), a non-blank character is entered in position [72] to indicate that the call is continued on the next statement. If a non-blank character is not entered in position [72] and a system delimiter (#) appears in position [14] of the next PR statement, it is assumed that the parameter list is complete for the current call and that the following PR statement starts a new call. |
| 72 | Continuation character – enter any non-blank character. |

Three kinds of entries are specified in a call and must appear in the following order: the CALL command, a load module name, and the fields (parameters) to be passed to the invoked routine. The specification of these entries must adhere to the following rules.

| Entry | Explanation | |
|---|---|---|
| Load module name | The entry immediately following the CALL command must be the name of a reusable module that is loaded into main storage by VISION:Builder before processing starts and receives control when the call is issued. The parameter list, as constructed by the user, is passed to this module (routine). Note that only one copy of a given module is loaded, regardless of how many CALL statements reference that module name. | |
| | OS/390 | The name must be the name of a load module accessible to VISION:Builder using the LNKLIST or a concatenation thereof or using a STEPLIB or JOBLIB DD statement. |
| | VSE | The named routine must be accessible using the PHASE sublibrary chain in the SEARCH parameter on the LIBDEF statement. |
| | CMS | The name must be the name of a TEXT file or a member of a TEXT library or a member in a LOADLIB defined in the GLOBAL LOADLIB command. |
| Parameter list (fields to be passed to the invoked routine) | In fixed format syntax, these parameters take the form qualifier#fieldname. Field names must follow VISION:Builder naming conventions. All VISION:Builder qualifiers except P (pattern) and L (tables) are permissible, including W and V, the working storage and linkage section qualifiers.<br><br>If no qualifier is specified, it is assumed that the name references a field in the new master file. Variable length fields cannot be specified as parameters in a CALL statement and implicit temporary fields can be used only if they have been referenced in a previous statement.<br><br>Character string constants and decimal constants are valid parameter specifications and have the following characteristics.<br><br>*C qualifier*: identifies a character string constant. Any character recognized by the hardware can be used in the constant except the system delimiter. The character string can be up to 31 characters in length including embedded blanks. Leading blanks are significant in character string constants. Trailing blanks cannot be specified as part of a character string constant because they are interpreted as spaces between parameters. | |

| Entry | Explanation |
|---|---|
| | *D qualifier*: identifies a decimal constant, which can be an integer or a number containing an explicit decimal point. Up to 15 digits can be specified. When a decimal constant is passed as a parameter in a CALL statement, the value is passed as an eight-byte, packed decimal number. No decimal points, commas, or other edit characters are passed; the receiving program must know the format of the field. Floating point notation cannot be used. |

Figure 18-10 on page 18-9 shows how character string and decimal constants might appear in a call.

## Return Code and Call Status Flags

Two flags, RETURNCD and CSTATUS, are available for you to determine the status of a call. RETURNCD values are set by the called routine, while CSTATUS values are set by VISION:Builder. Following any call, the contents of the return code flag field (RETURNCD) can be examined to determine the results of the call as set by the called routine. The RETURNCD flag is a four-byte, fixed point field. However, when it is included in reports, VISION:Builder edits it as though it were a two-byte fixed point field, so the output edit width is seven and the maximum value printable is 99,999. The value in this field is the value in general register 15 upon return to VISION:Builder from the called routine. This value can be set by Assembler programs by loading the register with the value. COBOL programs use the RETURN-CODE keyword.

When testing for a specific value in the return code flag field, you must know the possible settings, which depend on the called routine. Figure 18-11 illustrates the use of the return code flag.

```
CSAMPRET: PROC
; EXAMPLE OF CALL WITH RETURN CODE FLAG
;
CALL MODULE EPNAME USING T. FUNCTION W. IOAREA T. FLDN
   IF F. RETURNCD = 0   ;was CALL successful?
   ;
   ; process data from successful CALL
   ;
   ELSE ;
   ;
   ; handle error return from CALL
   ;
   END
END PROC
```

Figure 18-11 Return Code Flag

Additional call status information is obtained from the call status (CSTATUS) flag, which provides a method of determining in a VISION:Builder procedure if a call was suppressed by VISION:Builder and, if so, why. A call will be suppressed if a parameter specifies a field that is invalid or missing.

The call status flag is a four-byte character string, which can have the values shown in the following table.

| Value | Meaning |
|---|---|
| PMIS | Parameter missing |
| PINV | Parameter invalid |
| blank | Call successful |

The call status flag should be examined before using a value from the return code flag, because a suppressed call would not allow the called routine to set the return code. Figure 18-12 illustrates using a call status flag.

```
STATCHK: PROC
; example of CALL with CSTATUS code flag
;
CALL MODULE USERMOD USING T. FUNCTION T. FLDX
   IF F. CSTATUS = ' '   ;did CALL complete?
      IF F. RETURNCD = 0 ;was CALL successful?
      ;
      ; process data from successful CALL
      ;
      ELSE
      ;
      ; handle error return from CALL
      ;
      END
   ELSE
   ;
   ; CALL was not completed - examine CSTATUS flag for reason
   ;
   END
END PRC
```

Figure 18-12 Using the Call Status Flag

## Call Considerations

The considerations described in this section should be studied carefully when writing a call to an external program.

Through the use of calls, user I/O, and own-code, VISION:Builder allows you to call subroutines written in other languages such as COBOL, FORTRAN, and PL/I.

Unless specifically disabled via an APAR, VISION:Builder calls external programs with a valid IBM Language Environment enclave established. External programs compliant with the IBM Language Environment do not need any special interface considerations. The following language-specific sections discuss special considerations when an external program is not IBM Language Environment-compliant.

## Calling a VS COBOL II Subprogram

The following example illustrates how VISION:Builder can call a VS COBOL II subprogram. A key point to consider when calling a COBOL subprogram is that the COBOL environment must be created on or before the first call; usually this

environment should not be re-created (or initialized) during subsequent calls to the subprogram. If the subprogram is called many times, any calls after the first call would usually want to reuse the subprogram in its last-used state. This can be accomplished in VS COBOL II by the use of the COBOL run time library module named IGZERRE.

Figure 18-13 illustrates a COBOL subprogram that requires that the last-used state of the program be preserved. Note that GOBACK is used to return to the calling program. STOP RUN should not be used as unpredictable results will occur.

```
IDENTIFICATION DIVISION.
 PROGRAM-ID.                            COBSAMP.
 AUTHOR.                                VISION:Builder.
 DATE-WRITTEN.                          TODAY.
 DATE-COMPILED.                         TODAY.
*
*
 ENVIRONMENT DIVISION.
 CONFIGURATION SECTION.
 SOURCE-COMPUTER.                       IBM-370.
 OBJECT-COMPUTER.                       IBM-370.
*
*
 DATA DIVISION.
 WORKING-STORAGE SECTION.
*
 01  WORK-COUNT              PIC 9(5) COMP.
*
*
 LINKAGE SECTION.
*
 01  FUNC-CODE               PIC X(4).
*
 01  CALL-COUNT              PIC 9(5) COMP.
*
 01  RESULT-DATA             PIC X(20).
*
*
 PROCEDURE DIVISION USING FUNC-CODE CALL-COUNT RESULT-DATA.
*
 000-MAINLINE.
     IF FUNC-CODE = 'INIT'
*        PUT INITIALIZATION CODE HERE
         MOVE 0 TO WORK-COUNT
         GOBACK.

     IF FUNC-CODE = 'DOIT'
*        PUT PROCESSING FUNCTION CODE HERE
         ADD 1 TO WORK-COUNT
         MOVE WORK-COUNT TO CALL-COUNT
         ACCEPT RESULT-DATA FROM TIME
         GOBACK.
*
     IF FUNC-CODE = 'TERM'
*        PUT TERMINATION CODE HERE
         GOBACK.
*
```

Figure 18-13 Sample Call to COBOL II Subprogram, COBOL Code

Figure 18-14 illustrates the OS/390 JCL required to compile and link edit the COBOL subprogram and to run a VISION:Builder application to test the COBOL program.

```
  //  JOB ...
 //*
 //*--------------------------------------------------------------------
 //CLCOBOL  EXEC COB2VCL
 //*
 //* STEP 1: COMPILE AND LINKEDIT COBOL SUBPROGRAM.
 //*
 //COB2.SYSIN   DD DSN=your.source.library(COBSAMP),DISP=SHR
 //LKED.SYSIN   DD *
   NAME COBSAMP(R)
 //*
 //*--------------------------------------------------------------------
 //TESTRUN  EXEC PGM=MARKIV,REGION=2M,TIME=(,10),COND=(0,NE)
 //*
 //* STEP 2: TEST THE COBOL SUBPROGRAM.
 //*
 //STEPLIB  DD DSN=your.markiv.loadlib,DISP=SHR
 //         DD DSN=SYS1.COB2LIB,DISP=SHR             COBOL RUN-TIME LIB
 //         DD DSN=&&SYSLMOD,DISP=(OLD,DELETE)       FROM COBOL LINKEDIT
 //M4LIB    DD DSN=your.comlib.m4lib,DISP=SHR
 //M4LIST   DD SYSOUT=*
 //SYSOUT   DD SYSOUT=*
 //SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,5,,CONTIG)
 //SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,5,,CONTIG)
 //SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,5,,CONTIG)
 //M4REPO   DD UNIT=SYSDA,SPACE=(CYL,(1,1))
 //M4OLD    DD DSN=your.old.master,DISP=SHR
 //M4INPUT  DD DSN=your.markiv.program,DISP=SHR
 //
```

Figure 18-14 Sample Call to COBOL II Subprogram, JCL

Figure 18-15 depicts some VISION:Builder procedures that call the sample subprogram shown in Figure 18-13.

```
MAIN: PROC
   ;each master-file-record control procedure
FIRSTIME: FIELD C 1 INIT 'Y'
IGZFUNC1: FIELD F 4 INIT 1  ;function code to create
                            ;COBOL environment.
IGZFUNC2: FIELD F 4 INIT 2  ;function code to terminate
                            ;COBOL environment.
COUNT:  FIELD F 4 EDLEN 6
RESULT: FIELD C 20
;
   IF T. FIRSTIME = 'Y'
      CALL MODULE IGZERRE ,  ;create reusable, run-time,
           USING T. IGZFUNC1 ;COBOL environment.
      LET T. FIRSTIME = 'N'
   END
   ;
   CALL MODULE COBSAMP ,     ;call sample COBOL subprogram
        USING 'DOIT' T. COUNT T. RESULT
   CALL REPORT LISTINFO
END PROC
;
EOF: PROC TYPE EOF
   ;end-of-file control procedure
   ;
   CALL MODULE IGZERRE ,     ;terminate COBOL environment
        USING T. IGZFUNC2
END PROC
```

Figure 18-15 Sample Call to COBOL II Subprogram (Page 1 of 2)

```
LISTINFO: REPORT TYPE SUBROUTINE COLUMNS T.COUNT,  T.RESULT
          FORMAT HEADINGS NAME
          TITLE 'COBSAMP INFORMATION REPORT'
END REPORT

                                                          PAGE 1
               COBSAMP INFORMATION REPORT
---------------------------------------------------------------
  COUNT   RESULT
---------------------------------------------------------------
      1  15380074
      2  15380074
      3  15380074
      4  15380074
      5  15380074
      6  15380074
      7  15380074
      8  15380074
      9  15380075
     10  15380075
```

Figure 18-15 Sample Call to COBOL II Subprogram (Page 2 of 2)

Note the CALL command for module IGZERRE in the MAIN procedure. This CALL command creates the COBOL run-time environment and ensures that COBSAMP is treated as a subprogram. This call to IGZERRE using the initialization parameter should be made only once and a first time switch is used to limit the call to only the first time that the procedure MAIN is invoked. The procedure EOF contains a CALL command for module IGZERRE using the termination parameter that terminates the COBOL environment. Note that the times shown in the RESULT field of the report output give evidence that the COBOL environment persists between the calls to the COBSAMP subprogram.

See the *VS COBOL II Application Programming Guide* for a further discussion of the use of the IGZERRE module and other methods that can be employed to control the COBOL environment.

## COBOL/Assembler Interface Module

Because VISION:Builder enables all interrupts when passing control to a subroutine, you can ensure the validity of your data by trapping operating system interrupts in your subroutines. If you want to ignore operating system interrupts, a simple Assembler interface (Figure 18-16) can be linked with the subroutine and must be specified as the entry point of the generated module (that is, the routine in Figure 18-16 and the user subroutine).

This approach can be used as an alternative to the approach detailed in the previous section.

```
          To be LINKEDITED with COBOL program.
 ASMUSER   CSECT
           USING        *,15
           STM          0,15,SAVE              -- (save registers)
           CLI          SWITCH,C'Y'
           BNE          CALLUSER
           MVI          SWITCH,C'N'
           L            15,=V(XXXXXXXX)        -- (call COBOL interface)
           DROP         15
           BALR         14,15
```

Figure 18-16 Assembler Interface Routine for COBOL (Page 1 of 2)

```
          USING     *,14
          LM        0,15,SAVE             -- (restore registers)
          DROP      14
          USING     ASMUSER,15
CALLUSER  L         15,=V(USERPGM)        -- (call user routine USERPGM)
          SR        5,5
          ICM       5,8,=X'3n'            -- (see Note 2 below)
          SPM       5
          BR        15
SAVE      DC        16F'O'
SWITCH    DC        C'Y'
          END ASMUSER
```

Figure 18-16 Assembler Interface Routine for COBOL (Page 2 of 2)

**Note:** Replace XXXXXXXX with the module ILBOSTP0 for an OS/390 COBOL subroutine and with ILBDSET0 for a VSE COBOL subroutine.

## Disabling Interrupts

If you want all interrupts disabled, leave the ICM statement out completely, because the SR 5,5 sets the Program Mask to all zeros.

Of the hexadecimal value 3n, the 3 is the condition code that has no meaning in the PSW. The 3 was arbitrarily chosen and can be replaced with any hexadecimal number. The n represents the Program Mask of the PSW. The contents of this last half-byte determines the handling of the following interrupts:

- Bit 1: Fixed Point Overflow
- Bit 2: Decimal Overflow
- Bit 3: Exponent Overflow
- Bit 4: Significance.

| Use this value in a bit | To do this |
|---|---|
| 0 | Disable the interrupt and a 1 enables the interrupt. |
| X'30' | Disable all interrupts. |
| X'31'<br>.<br>. | Enable Fixed Point Overflow interrupt and disables all others. |
| X'3F' | Enable all interrupts. |

If this interface is not used, the COBOL program can abend when called by VISION:Builder if the above noted data exceptions occur.

## The Fortran/Assembler Interface Module

Figure 18-17 shows an Assembler interface subroutine for FORTRAN.

```
         To be LINKEDITED with FORTRAN program.
ASMUSER  CSECT
         USING     *,15
         STM       0,15,SAVE              -- (save registers)
         L         15,=V(IBCOM)           -- (call FORTRAN interface)
         DROP      15
         BALR      14,64(15)
         USING     *,14
         LM        0,15,SAVE              -- (restore registers)
         DROP      14
         USING     ASMUSER,15
CALLUSER L         15,=V(USERPGM)         -- (call user routine USERPGM)
         SR        5,5
         ICM       5,8,=X'30'
         SPM       5
         BR        15
SAVE     DS        16F"0"
         END       ASMUSER

 If this interface is not used, the non-assembler routines may ABEND when
 called by VISION:Builder. This is because the FORTRAN environment has
 not been established.
```

Figure 18-17 Assembler Interface Routine for FORTRAN

## The PL/I Procedure Statement

When calling PL/I routines, do not overlook the role of the PL/I environment. Before a PL/I routine executes, you must establish the environment and, after the PL/I routine is done executing, you must cleanup the environment. If you repeatedly call a PL/I routine, as in a loop, establish the environment once before the first call and cleanup the environment once after the routine returns for the last time. This can save CPU time and disk I/O that would be used to establish the environment.

Figure 18-18 shows the BITRTN Assembler routine. This routine handles three major calling conditions:

- First call to the PL/I routine.
- Cleanup call to the PL/I routine.
- Other intermediate calls.

```
*                                  ********************
*                                  *****  MODIFY  *****
*                                  ********************
BITRTN   CSECT                     IN THIS MODULE NAME VISION:TWO CALL;
*                                  IT MUST BE UNIQUE FOR EACH PL/I SUBRTN.
         ENTRY    PLICALLA
PLICALLA DS       0H
         USING    *,15
         LTR      1,1
         BZ       EXIT
```

Figure 18-18 Assembler Routine BITRTN (Page 1 of 2)

```
*******************************************************************************
*    THE CODE FROM HERE TO LABEL "INIT" IS EXECUTED ONLY ON THE               *
*    FIRST CALL TO BITRTN. IT SAVES REGISTERS IN THE SAVE AREA                *
*    SUPPLIED BY VISION:TWO, AND CHAINS IN TWO LOWER LEVEL SAVE-              *
*    AREAS, S1 AND S2. IT SUPPLIES THE S2 SAVEAREA IN CALLS TO                *
*    THE PL/I ROUTINE ALLOWING PL/I TO CHAIN BACK TO THE SAVE-                *
*    AREAS CONTAINING THE PL/I ENVIRONMENT. THIS INCLUDES THE                 *
*    CHAIN CONTAINING THE PL/I DSA FOR OUR CONTROL BLOCK.                     *
*    THE PL/I INTERLANGUAGE ROUTINE CHAINS 3 SAVEAREAS ABOVE THE              *
*    S1 SAVEAREA INSTEAD OF PUTTING THEM ABOVE THE SAVEAREA                   *
*    SUPPLIED BY THE SYSTEM TO VISION:TWO.                                    *
*******************************************************************************
         CLI      ENVIRON,X'FF'              BYPASS ONE TIME CHECK
         BZ       INIT
         MVI      ENVIRON,X'FF'              CLOSE GATE
         STM      14,12,12(13)               SAVE CALLER'S REGS
         ST       13,S1+4                    SAVE CALLER'S REG 13
         ST       13,S2+72                   SAVE FOR ENVIRON RESTORE
         LA       11,S1                      GET OUR SAVEAREA
         ST       11,8(13)                   AND CHAIN BACK
         LA       14,RET                     SET ONE TIME TRAP
INIT     DS       0H                         SKIP POINT
         LA       13,S2                      SAVEAREA FOR PLI SUBRTN
*                                   ********************
*                                   *****  MODIFY  *****
*                                   ********************
         L        15,=V(BITSUB)              NAME OF PLI ROUTINE
         BR       15
EXIT     L        15,=V(CLEANUP)             ASSM. MODULE THAT CLEANS
*                                            UP PLI ENVIRONMENT ON
*                                            LAST CALL (NO PARMS ARE
*                                            PASSED ON LAST CALL).
         BR       15
RET      L        13,72(13)
         ST       15,16(13)                  SAVE PLI RETURN CODE
         LM       14,12,12(13)
         BR       14
S1       DC       A(0),A(0),A(S2),15F'0'
S2       DC       A(0),A(S1),A(0),16F'0'
         DC       V(PLIENV)
ENVIRON  DC       XL1'00'
         END      BITRTN
```

Figure 18-18 Assembler Routine BITRTN (Page 2 of 2)

The first call to the PL/I routine is issued in the first loop of the MAIN procedure. The last call is issued in the EOF procedure by having no parameters. Its parameter list pointer is logically zero. All intermediate calls are just like the first call, except that the ENVIRON field in the BITRTN Assembler routine is set to X'FF'.

To complete the interface are the two PL/I routines BITSUB and PLIENV (see Figure 18-19 and Figure 18-20) and the Assembler routine CLEANUP (see Figure 18-21). BITSUB (shown in Figure 18-19) is the PL/I routine that makes the bit manipulation possible. The most important things for you to notice about the BITSUB code is the way the parameters are received from the application procedure MAIN. Pay special attention to the DCLs (declares) in BITSUB and the FIELD commands in the MAIN procedure. PLIENV is a dummy PL/I routine that is called when no parameters are passed. It is used to bring in the established PL/I environment so it can be cleaned up. The CLEANUP Assembler routine does all the work to cleanup the environment.

```
BITSUB: PROC  (I1,I2,I3,I4)  OPTIONS  (COBOL,NOMAP)  REORDER;
/*
/*  PURPOSE:  TO CONVERT FIELD, VISION:TWO, TO A CHARACTER
/*      STRING OF THE FORMAT:  " 'XXX...'B " WHERE X IS ZERO OR ONE
/*      (for example, '01010'B). VISION:TWO CAN THEN PROCESS THE STRING WITH
/*      PARTIAL FIELDING TO SIMULATE BIT MANIPULATION.
/*
/*  NOTE:  BECAUSE OF THE WAY PLI PASSES PARAMETERS. THE PROGRAM
/*      ACCEPTS "DUMMY" NUMERIC PARAMETERS (II,12013,14) THEN OVERLAYS
```

Figure 18-19 BITSUB PL/I Routine

```
/*     THEM WITH BASED VARIABLES HAVING THE CORRECT DATA TYPE.
*/
  DCL  BITSTRG BIT(32)  BASED  (P1), /* BIT STRING FROM STERLING ANSWER/2 /*
       OFFSET  FIXED DEC(15) BASED (P2),  /* OFFSET INTO BIT STRING AT
                                          WHICH TO START PROCESSING       /*
       NUMBITS FIXED DEC(15) BASED (P3),  /* NUM OF BITS TO PROCESS       /*
       RESULT CHAR(255)BASED(P4);  /* OUTPUT FIELD RETURNED TO A/2        /*
  DCL (P1,P2,P3,P4)  POINTER,
      (I1,I2,I3,I4),
      (SUBSTR,ADDR)  BUILTIN;

  P1 = ADDR(I1);
  P2 = ADDR(I2);
  P3 = ADDR(I3);
  P4 = ADDR(I4);
/*   CONVERT INPUT FIELD TO OUTPUT CHARACTER STRING  /*
  SUBSTR(RESULT,1,NUMBITS+3)  =
       '''' || SUBSTR(BITSTRG,OFFSET,NUMBITS)  ||  '''B';
END;
```

Figure 18-19 BITSUB PL/I Routine

The JCL for the interface is shown in Figure 18-22. Refer to that figure and follow the routines below (Figure 18-20 and Figure 18-21) to interface PL/I with VISION:Builder.

```
PLIENV:  PROC  OPTIONS(MAIN)    REORDER;
/*   SKELETON  PLI  MAIN  MODULE  USED  TO  BRING  IN  PLI  ENVIRONMENT.
RETURN;
END        PLIENV;
```

Figure 18-20 PLIENV PL/I Routine

```
CLEANUP  CSECT
         USING   *,ENTER
         STM     R0,ENTER,SAVEENT     SAVE REGISTERS|
         DROP    ENTER
*******************************************************************************
*   CLEANUP CLOSES PLI ENVIRON. PLI ADDS SAVEAREAS IN ZYGO-LINGUAL
*      CONTROL LIST (ZCTL) TO SET ENVIRON. PLI ADDS 3 SAVEAREAS ABOVE
*      CALLER SAVEAREA, BETWEEN CALLER'S CALLER AND CALLER'S CALLER'S
*      CALLER. TOP CHAIN IS MODIFIED TO POINT TO PLI INIT ROUTINE TO
*      CLOSE ENVIRONMENT. REG14 AND CHAIN DATA IS SAVED IN 4-BYTE
*      GHOST SAVEAREA IN ZCTL.
*   THE ROUTINE COPIES GHOST SAVEAREA, ZAPS ZCTL TO PASS CONTROL TO
*      LABEL-RETURN AFTER ENVIRON IS CLOSED. POSITIONS AT SAVEAREA BELOW
*      ADDED SAVEAREAS, AND RETURNS. PLI ROUTINES CLOSE ENVIRON & PASS
*      CONTROL TO LABEL-RETURN, THAT REMOVES HOOKS RETURNING TO CALLER.
*******************************************************************************
          BALR  BASE,R0              GET BASE
          USING *,BASE               GET ADDRESSIBILITY
          L     WORK1,=V(IBMBILC1)   GET ADDRESS OF CONTROL BLOCK
          LTR   WORK1,WORK1          DOES THIS MODULE EXIST?
          BZ    CALLRETN             NO MORE PROCESSING
          L     ZCTL,0(WORK1)        GET ADDRESS OF ZCTL
          LTR   ZCTL,ZCTL            IS THERE A SAVEAREA?
          BZ    CALLRETN             NO ENVIRONMENT OPEN, RETURN
          CLI   4(WORK1),X'00'       IS BLOCK OPEN?
          BE    CALLRETN             BRANCH IF BLOCK CLOSED.
          MVC   GHOST(16),128(ZCTL)  GET GHOST SAVEAREA
          LA    WORK2,RETURN         GET NEW ADDRESS FOR SYSTEM RETURN
          ST    WORK2,140(ZCTL)      SAVE NEW REGISTER 14
          L     SAVE,48(ZCTL)        POINT TO SAVEAREA
          LM    LINK,BASE,12(SAVE)   RESTORE REGISTERS
          BR    LINK
```

Figure 18-21 CLEANUP Assembler Routine (Page 1 of 2)

```
*******************************************************************************
*   LABEL RETURN GETS CONTROL WHEN ENVIRONMENT IS CLOSED. GHOST SAVE
*     AREA WAS SAVED BY PREVIOUS CODE AND OUR ADDRESS IS PATCHED IN
*     GHOST SAVEAREA. WHEN WE GET CONTROL, REG 13 POINTS TO CALLER'S
*     CALLER SAVEAREA. CODE RESTORES REG 14 TO SAVEAREA AND RETURNS.
*******************************************************************************
RETURN     BALR  BASE,R0
           USING *,BASE
           L     LINK,GHOST14
           ST    LINK,12(SAVE)
CALLRETN   BALR  BASE,R0
           USING *,BASE
           LM    R0,ENTER,SAVEENT
           BR    LINK
R0         EQU   0
ZCTL       EQU   9
WORK1      EQU   10
WORK2      EQU   11
BASE       EQU   12
SAVE       EQU   13
LINK       EQU   14
ENTER      EQU   15
SAVEENT    DS  18F
GHOST      DS   3F
GHOST14    DS    F
           END
```

Figure 18-21 CLEANUP Assembler Routine (Page 2 of 2)

## A Cookbook Approach to Implementation

- Code and compile the PL/I programs; pass parameters as shown in example below (see steps 1 and 2 in the JCL in Figure 18-22).
- Code and assemble the interface and cleanup routines; there is one such interface module for each PL/I subroutine (see steps 3 and 4 in the JCL in Figure 18-22). The two lines of code in the interface routine that need changing are marked by the word modify in the source.
- Link edit the PL/I and Assembler routines (see step 5 in the JCL in Figure 18-22).

```
  //  JOB ...
//*
//*--------------------------------------------------------------------
//CLPLIENV EXEC PLIXC,
//   PARM.PLI='AG,A,MAR(2,72,1),NOMAP,NEST,OF,OPT(TIME),X'
//*
//* STEP 1: COMPILE SKELETON PLI MAIN MODULE.
//*
//PLI.SYSIN    DD DSN=your.source.library(PLIENV),DISP=SHR
//*
//*--------------------------------------------------------------------
//CLBITSUB EXEC PLIXC,
//   PARM.PLI='AG,A,MAR(2,72,1),NOMAP,NEST,OF,OPT(TIME),X'
//*
//* STEP 2: COMPILE PLI SUBPROGRAM BITSUB.
//*
//PLI.SYSIN    DD DSN=your.source.library(BITSUB),DISP=SHR
//*
```

Figure 18-22 JCL to Interface PL/I to VISION:Builder (Page 1 of 2)

```
//*-------------------------------------------------------------------
//ASBITRTN EXEC ASMFC
//*
//* STEP 3: ASSEMBLE INTERFACE MODULE BITRTN.
//*
//ASM.SYSIN    DD DSN=your.source.library(BITRTN),DISP=SHR
//*
//*-------------------------------------------------------------------
//ASMCLEAN EXEC ASMFC
//*
//* STEP 4: ASSEMBLE CODE TO CLEAN PLI ENVIRONMENT AFTER CALLS
//*         TO PLI ARE DONE.
//*
//ASM.SYSIN    DD DSN=your.source.library(CLEANUP),DISP=SHR
//*
//*-------------------------------------------------------------------
//LKBITRTN EXEC PGM=IEWL,REGION=1M,PARM='LET,LIST,XREF'
//*
//* STEP 5: LINKEDIT SKELETON PLI MAIN MODULE, PLI SUBPROGRAM BITSUB,
//*         ASSEMBLER INTERFACE MODULE AND ASSEMBLER CLEANUP
//*         MODULE INTO AN EXECUTABLE LOAD MODULE.
//*
//ASMOBJ   DD DSN=&&SYSLIN,DISP=(OLD,DELETE)       ASSEMBLER OBJECT
//PLIOBJ   DD DSN=&&LOADSET,DISP=(OLD,DELETE)      PL/I COMPILER OBJECT
//SYSUT1   DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSLIB   DD DSN=SYS1.PLIBASE,DISP=SHR
//         DD DSN=SYS1.SIBMBASE,DISP=SHR
//SYSLMOD  DD DSN=&&LOADLIB,DISP=(NEW,PASS),       PASS TO MARK IV STEP
//            UNIT=SYSDA,SPACE=(CYL,(1,1,10))
//SYSPRINT DD SYSOUT=*
//SYSLIN   DD *
  INCLUDE ASMOBJ
  REPLACE PLICALLA
  INCLUDE PLIOBJ
  ENTRY PLICALLA
  NAME BITRTN(R)
//*
//*-------------------------------------------------------------------
//TESTRUN  EXEC PGM=MARKIV,REGION=1M,TIME=(,10)
//*
//* STEP 6: TEST THE INTERFACE WITH VISION:Builder.
//*
//STEPLIB  DD DSN=your.markiv.loadlib,DISP=SHR
//         DD DSN=&&LOADLIB,DISP=(OLD,DELETE)         FROM LKBITRTN STEP
//M4LIB    DD DSN=your.comlib.m4lib,DISP=SHR
//M4LIST   DD SYSOUT=*
//SYSOUT   DD SYSOUT=*
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,5,,CONTIG)
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,5,,CONTIG)
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,5,,CONTIG)
//M4REPO   DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//M4OLD    DD DSN=your.old.master,DISP=SHR
//M4INPUT  DD DSN=your.markiv.program,DISP=SHR
//
```

Figure 18-22 JCL to Interface PL/I to VISION:Builder (Page 2 of 2)

## VISION:Builder Call Execution

When VISION:Builder executes a CALL statement, a parameter list is constructed with the address of each parameter (if any are specified) appearing in the list in the order specified in the CALL statement. If a specified parameter is invalid or missing, the CALL statement is not executed, the call status flag is set, and the next VISION:Builder statement is executed. Following standard call conventions, the last address in the list has a hex '80' in the high order byte to indicate the end of the

list. Also adhering to standard programming practices, you cannot assume that the address of a parameter remains the same from call to call. When a CALL statement is generated, the following call conventions are observed by VISION:Builder:

- The entry point address of the called routine is placed in register 15.
- The return point to VISION:Builder is placed in register 14.
- The address of a save area of 18 fullwords is placed in register 13.
- The address of the parameter list is placed in register 1.
- Control is passed to the named module and, upon return, the contents of register 15 are placed in the return code flag.

The following points should also be observed when specifying CALL statements.

- Calls do not affect the validity or invalidity of a field specified as a parameter. A field that is valid prior to a call is assumed valid by VISION:Builder following the call.
- Fields defined in the linkage section can be altered by VISION:Builder procedure processing if the modification is permitted by the program that invoked VISION:Builder (for example, the field does not reside in a protected area of storage).
- The FREESIZE keyword on the CONTROL command (FREESIZE parameter on the RP statement in fixed format syntax) can be used to free a specified amount of main storage to satisfy the dynamic storage requirements of called programs during the processing phase of VISION:Builder.
- Care should be taken by the called routine not to modify fields specified as constants, count fields, or key fields when passed as parameters.
- Calls should not be made that results in an I/O access to any file being automatically processed concurrently by VISION:Builder.
- Program interrupts in a called routine are handled in the same manner as they are in own-code routines (see the *VISION:Builder for OS/390 Environment Guide* (OS/390 users) or the *VISION:Builder for VSE Environment Guide* (VSE users)).

Automatic looping in a call can occur when a lower level segment type is referenced as a parameter within the call. All of the rules for automatic looping apply in this case (see Chapter 8, *Data Structure Handling*). The call is likely to be executed multiple times, once for each lower level segment occurrence.

## Calls to IMS

Calls to IMS can be made from VISION:Builder procedure processing by coding a call to module ASMTDLI. The entry point, MARKDLI or MARKDLIX, should be used when executing the IMS region controller DFSRRC00. MARKDLIX is used when the IMS checkpoint/restart facility is being used. (See the *VISION:Builder for*

*OS/390 Environment Guide* (OS/390 users) or the *VISION:Builder for VSE Environment Guide* (VSE users) for further information on the use of checkpoint/restart with IMS.)

The input parameter list must contain the addresses of the PCBs required for the run. Linkage section entries can be used to define any PCB passed to VISION:Builder by the IMS region controller. Linkage section definitions are only required if the application procedures need to reference the PCB or its contents.

IMS has certain dynamic storage requirements during the processing phase of VISION:Builder. When using the MARKDLI or MARKDLIX entry point, VISION:Builder frees an amount of storage that is usually sufficient for IMS purposes. This value may have to be adjusted to accommodate IMS requirements. Regardless of the entry name, the storage freed by VISION:Builder can be specified using the FREESIZE keyword on the CONTROL command or using the FREESIZE parameter on the RP statement in fixed format VISION:Builder syntax.

A unique PCB is required for each IMS database that is accessed as a VISION:Builder logical file. This PCB contains the processing options required to handle the file and is a subset of the PSB specified at run time. Calls to IMS require the use of a PCB, which must be a different PCB than any used by VISION:Builder. Loss of position within a database and erroneous processing can result from an attempt to share PCBs.

While a logical record is being accessed by VISION:Builder, it should be considered part of the VISION:Builder internal environment. Altering the structure of this record could cause incorrect processing. As an example, assume that VISION:Builder is inserting❹ a new segment into an IMS database record. The segment is logically inserted in the record held in main storage, but is not physically added to the database until after procedure processing. In a procedure, a call could be made to IMS to physically insert the same segment. When VISION:Builder attempts to physically insert the segment at the end of procedure processing, the action is rejected by IMS.

In general, you should try to avoid duplicating VISION:Builder processing on database records. Updating, for example, should be performed entirely by VISION:Builder❹ or entirely through calls. Similarly, it is better to retrieve segments entirely through calls or entirely with the automatic facility of VISION:Builder.

## Calls to DL/I VSE Environment

Calls to DL/I can be made from VISION:Builder procedure processing in the VSE environment. The name MARKDLI should be used as the application program name when executing the DL/I controller DLZRRC00. The input parameter list must contain the addresses of the PCBs required for the run. Linkage section entries define any PCB passed to VISION:Builder by the DL/I VSE controller. If DL/I files are not used, the name MARKGSI or MARKGSS can be used.

DL/I has certain dynamic storage requirements during the processing phase of VISION:Builder. When MARKDLI is entered, the amount of storage freed is determined by the MINCORE value in M4PARAMS. This value may have to be

adjusted to accommodate DL/I requirements. Regardless of the entry name, the storage freed by VISION:Builder can be specified using the FREESIZE keyword on the CONTROL command or using the FREESIZE parameter on the RP statement in fixed format syntax.

## Calls to IBM Language Environment Services

Language Environment (LE) from IBM provides a number of callable services for programming language products. VISION:Builder programs can call any LE service. VISION:Builder will automatically call selected LE date and time services whenever any conversion from or to Lilian Date type fields is needed. User calls to LE services are called just like any other external program as described in Calls from VISION:Builder to an External Program on page 18-8 of this chapter. These calls can be coded as ASL CALL commands or as fixed format syntax PR statements with the CALL operation. It is the responsibility of the user to provide the correct parameters for each LE service called. These parameters are described in the IBM manual *Language Environment for MVS and VM Programming Reference*.

The ASL free format syntax of VISION:Builder includes special facilities to recognize calls to LE date and time services. When these services are called using the ASL CALL command, VISION:Builder will automatically convert any parameters required to be a length-prefixed character string (VSTRING) to the appropriate format. The date and time services that ASL recognizes for parameter conversion are:

| | |
|---|---|
| CEEDATE | Converts dates in the Lilian format to character values |
| CEEDATM | Converts number of seconds to character timestamp |
| CEEDAYS | Converts character date values to the Lilian format |
| CEDYWK | Provides day of week calculation |
| CEEGMT | Gets current Greenwich Mean Time (date and time) |
| CEEGMTO | Gets difference between Greenwich Mean Time and local time |
| CEEISEC | Converts binary year, month, day, hour, minute, second, and millisecond to a number representing the number of seconds since 00:00:00 15 October 1582 |
| CEELOCT | Gets current date and time |
| CEEQCEN | Queries the century window |
| CEESCEN | Sets the century window |

| | |
|---|---|
| CEESECI | Converts a number representing the number of seconds since 00:00:00 15 October 1582 to seven separate binary integers representing year, month, day, hour, minute, second, and millisecond. |
| CEESECS | Converts character timestamps (a date and time) to number of seconds since 00:00:00 15 October 1582 |
| CEEUTC | Same as CEEGMT |

When the PR statement in fixed format syntax is used, there is no special recognition of the above LE date and time services. It is the responsibility of the user to provide any parameters expected to be passed as a length-prefixed character string (VSTRING) as a VISION:Builder variable length (V-type) field. The following figures show examples of calls to LE services using both the ASL free format syntax and the PR statement fixed format syntax.

```
IBMLERTN: PROC
;
LILDATE:  FIELD D 4
FEEDBACK: FIELD C 12
WEEKDAY:  FIELD C 80
;
; convert birth date to a Lilian date using picture 'YYYY-MM-DD'
;
CALL CEEDAYS USING BRTHDATE, 'YYYY-MM-DD',
     T.LILDATE, T.FEEDBACK
;
; convert Lilian date to a format showing day of the week
;
CALL CEEDATE USING T.LILDATE, 'Wwwwwwwwwz, Mmm DD, YYYY',
     T.WEEKDAY, T.FEEDBACK
;
END PROC
```

Figure 18-23 ASL CALL to IBM LE Services

```
                                          ****************************
                                          *  REQUEST NAME - REQUEST1  *
                                          *  INPUT STREAM REQUEST     *
                                          ****************************
-------------------------------------------------------------------------------------------------------------------
STMT   FIELD     FIELD FLD DEC OUTPT EDIT      INITIAL          LINE 1 OF       LINE 2 OF
TYPE   NAME      LNGTH TYP PLC EDIT  LGTH       VALUE         COLUMN HEADING   COLUMN HEADING
-------------------------------------------------------------------------------------------------------------------
(TF) (LILDATE ) (  4) (D)
(TF) (FEEDBACK) ( 12) (C)
(TF) (WEEKDAY ) ( 80) (C)
(TF) (TEMP___3) ( 24) (V)
(TF) (TEMP___2) ( 10) (V)               (YYYY-MM-DD     )
(TF) (TEMP___1) ( 10) (V)
-------------------------------------------------------------------------------------------------------------------
STMT SEQ  LOG CON .....OPERAND-A..... OPER  .............OPERAND-B.............  ......RESULT...... PARTIAL FIELD
TYPE NO. LEV CTR QLF FIELD   SEG-LVL  ATN   QLF FIELD   SEG-LVL  (OR CONSTANT)   QLF FIELD  SEG-LVL  STRT NUMB OPR
-------------------------------------------------------------------------------------------------------------------
(PR) (101)                            (R )     (BRTHDATE) 001-1                  (T,TEMP___1)
(PR) (103)  (#)      (CALL CEEDAYS  T#TEMP___1 T#TEMP___2 T#LILDATE           ) (X)
(PR) (104)  (#)      (T#FEEDBACK                                              )
(PR) (105)                            (R )    (C,Wwwwwwwwwz, Mmm DD, YYYY      ) (T,TEMP___3)
(PR) (107)  (#)      (CALL CEEDATE  T#LILDATE T#TEMP___3 T#WEEKDAY            ) (X)
(PR) (108)  (#)      (T#FEEDBACK                                              )
```

Figure 18-24 Fixed Format Syntax

Note that with the fixed format syntax, the programmer must ensure that the parameters required to be length-prefixed character string (VSTRING) parameters as specified in the LE documentation must be placed into VISION:Builder temporary V-type fields by user coding prior to calling the LE service. Note also that V-type fields referenced as parameters in a CALL statement must be Temporary V-type fields. V-type fields defined within a File Definition are not acceptable as parameters on a CALL statement.

Chapter

# 19 Cataloging

The common library contains a variety of item types that have been validated and cataloged to the library by VISION:Builder. Each type of item, with the exception of run data groups (see Run Data Groups on page 19-4), has been introduced in a previous chapter.

- File definitions are described in Chapter 3, *Defining Data*.
- Tables and arrays are described in Chapter 12, *Tables* and Chapter 13, *Arrays*.
- Transaction definitions and transaction groups are described in Chapter 10, *Transactions* and Chapter 15, *Advanced Transaction Processing*.❹
- Procedures, reports, and subfiles are described in Chapter 5, *The Application Cycle*.
- Requests are described in Chapter 11, *Requests*.

This section is intended to round out your knowledge of cataloging items to the library and of the methods used to update and maintain the cataloged items.

Refer to the *VISION:Builder for OS/390 Environment Guide* (OS/390 users) or the *VISION:Builder for VSE Environment Guide* (VSE users) and the *VISION:Builder Reference Summary* for details on cataloging and maintaining items in the common library. These specifications must be entered using the VISION:Workbench for ISPF component of VISION:Builder or any standard text editor. At this time, specifications for cataloging and maintaining items in the common library cannot be entered using VISION:Workbench for DOS.

## Definitions

There are a number of different types of definitions that can be cataloged to the common library. They include a file definition, table definition, array definition, transaction group definition,❹ and run data group.

Each of these item types is cataloged to the common library during a VISION:Builder definition/maintenance run.

Once a definition is validated and cataloged, you can list any item. The listing can consist of only the names of a specified type currently in the library or, optionally, can include a detailed glossary of selected items or all items of a specified type. These operations are performed in a VISION:Builder definition/maintenance run. You specify the operations you want performed and the items to which you want these operations to apply on a LISTLIB NAMES or LISTLIB GLOSSARY command.

You can retrieve any definition back into its fixed format source statement format. This retrieval operation is performed during a source statement retrieval run using the RETRIEVE command. These statements can be imported into VISION:Workbench for DOS for ongoing maintenance or you can use whatever editing tools you have available to perform source level maintenance on these items.

For details on cataloging, listing, and source statement retrieval of a definition, refer to the *VISION:Builder for OS/390 Environment Guide* (OS/390 users) or the *VISION:Builder for VSE Environment Guide* (VSE users) and *VISION:Builder Reference Summary*.

## Instream File Definitions

Chapter 3, *Defining Data* introduced the concept of a definition/maintenance run in which definitions are cataloged in a common library prior to an application run as introduced in Chapter 5, *The Application Cycle*. An alternative approach, which may be useful in some circumstances, is to include the file definition source statements instream with the application source statements. Even more useful in many instances is to include file definition source statements instream with the application source statements that temporarily modify or override parts of the cataloged file definition. In this way, a file definition can be modified temporarily in memory as required for a single application without the modifications being cataloged in the common library where they would affect all other applications using the base file definition.

The source statements for an instream definition must be placed in the Run Control command group before any catalog control statements (CATALOG or TRACK commands) or the first procedure or request of the application. See the *VISION:Builder Reference Summary* for a detailed description of the statement order requirements. Note that only file definition statements for external files can be included instream with an application. Table definitions, transaction definitions, and so on, or definitions for working storage and linkage sections cannot be included instream.

Source statements for an instream definition can be included within ASL free form syntax in one of the following ways:

- Definitions that exist in another file or PDS member can easily be inserted into the ASL command stream using the ASL: COPY command with the FIXED keyword. Refer to the *VISION:Builder ASL Reference Guide* for more information.
- Definition source can be included in the ASL command stream by wrapping the definition source with ;;BEGASL and ;;ENDASL.

Figure 19-1 displays a portion of an application source program containing an instream file definition. The statements between ;;ENDASL and ;;BEGASL are instream file definition statements. Note that a glossary for the file definition is listed following the instream file definitions statements. The glossary listing can be suppressed with the appropriate specifications on the RP LISTCNTL statement in fixed format or by cataloging or overriding the cataloged definition with the No Glossary specification on the FD statement in the definition. See the *VISION:Builder Reference Summary* for detailed information about the RP LISTCNTL parameter specifications.

```
CONTROL  SYSDATE 061402   LISTGEN
FILE MASTER INPUT SAMPLE
FILE REPORT
;
;;ENDASL
SAMPLE  FD        R F  80
SAMPLE  LSROOT      1
SAMPLE  L0NEWKEY1         1  4F1
SAMPLE  L0NEWFLDA         5 16C
SAMPLE  L0NEWFLDB        21 32C
SAMPLE  L0DUEDATE        53  6C
SAMPLE  L0PRINCIPL       59  8P  2
;;BEGASL
;
MAIN: PROC
   ;each master-file-record control procedure
COUNTER: FIELD F 4 INIT 0
   ;
  IF N.NEWFLDA EQ 'ACME INC. '
     LET N.NEWFLDB = 'REPORTED BY SAMPLE PROCEDURE'
     LET T.COUNTER = T.COUNTER + 1
       IF T.COUNTER GE 5
          LET PF(F.EOF 1 1) = 'E'
       ELSE
          CALL REPORT SAMPREP1
       END
  END
END PROC
;
SAMPREP1: REPORT TYPE SUBROUTINE,
               COLUMNS  N.NEWFLDA  N.NEWFLDB  N.DUEDATE
               FORMAT HEADINGS NAME
END REPORT
```

Figure 19-1 Application with Instream File Definition

# Run Data Groups

A run data group is a collection of VISION:Builder fixed format run control statements for an application that are cataloged on the common library. These run data groups are available to be used as the run control basis for applications without having to code all of the detail run control parameters for the application. Each application using a cataloged run data group would have the identical input/output file configuration, run time parameters, and so on.

Refer to the *VISION:Builder Reference Summary* for detail specifications on run data groups and cataloging, specifically, the Technical Notes chapter.

An ASL Run Content command group could alternatively be included using the COPY command.

Figure 19-2 is an example of the fixed format input statements to define a simple run data group.

```
-----------------------------------------------------------------------------------------------------------------------------------
  RUN     STMT    FILE    O N S D U T R A B S D S L  RB  R S M R S            START            END              SEQUENCE   LIST
  NAME    TYPE    NAME    L E E I P R E U U R L C S  PL  J O O F S            SEARCH          SEARCH            73 TO 80    SEQ
                          D W Q R D N P D F T M N T  TK  T P P O R
-----------------------------------------------------------------------------------------------------------------------------------
(CAT_RDG ) (RC)                                                                                                                1

** CLSAA00  TYPE 0  DEFINITION RDG1     IS BEING PROCESSED.                                                                  *****
-----------------------------------------------------------------------------------------------------------------------------------
  GROUP   STMT         DEL?  GLOSSARY                                                                                SEQUENCE   LIST
  NAME    TYPE                                                                                                        73 TO 80    SEQ
-----------------------------------------------------------------------------------------------------------------------------------
(RDG1    ) (DB)          (R)      (N)                                                                                           2
-----------------------------------------------------------------------------------------------------------------------------------
  GROUP   STMT  LOGICAL  LBL  DTF/DD      KEY       KEY       KEY      QUALIFIER   USER I/O  ARRAY       PASSWORD/  SEQUENCE   LIST
  NAME    TYPE  FILENAME       NAME     FIELD-1   FIELD-2   FIELD-3   ICF CRD DBI   MODULE   QUAL        AUTH.-ID   73 TO 80    SEQ
-----------------------------------------------------------------------------------------------------------------------------------
(RDG1    ) (RF)  (CUSTOMER) (S) (M4OLD   )                                                                                      3
(RDG1    ) (RF)  (ITEM    ) (S) (M4CORD1 )  (ITEMORD )                    (O)                                                   4
(RDG1    ) (RF)  (SUBFILE1) (S) (M4SUBF1 )                                                                                      5
-----------------------------------------------------------------------------------------------------------------------------------
  GROUP   STMT CURRENT DEL?  DTF/DD    ALIAS   ALIAS LOGICAL                                                          SEQUENCE   LIST
  NAME    TYPE  NAME          NAME     NAME    TYPE   UNIT                                                            73 TO 80    SEQ
-----------------------------------------------------------------------------------------------------------------------------------
(RDG1    ) (LB) (ITMPRICE)     (M4OLD   ) (PRICE   ) (F)                                                                        6
(RDG1    ) (LB) (CUSTNO  )     (M4CORD1 ) (BOCUSTNO) (F)                                                                        7
(RDG1    ) (LB) (ORDERNO )     (M4CORD1 ) (ITORDNO ) (F)                                                                        8
(RDG1    ) (LB) (QTYBKORD)     (M4CORD1 ) (ITQTYBKO) (F)                                                                        9
-----------------------------------------------------------------------------------------------------------------------------------
  GROUP   STMT  PARAMETER                                                                                             SEQUENCE   LIST
  NAME    TYPE    NAME                                  PARAMETER VALUE                                               73 TO 80    SEQ
-----------------------------------------------------------------------------------------------------------------------------------
(RDG1    ) (RP)  (FREESIZE)        (100K                                                  )                                     10
** CLSVN20  TYPE 0  THE DEFINITION IS VALID AND HAS BEEN CATALOGED.                                                            *****
```

Figure 19-2 Example of Cataloging a Run Data Group

To use a run data group, the run data group name is specified in place of the file definition name for the primary file (M4OLD). Figure 19-3 displays a portion of the fixed format input statements using the sample run data group cataloged in Figure 19-2.

Note that statements with List Seq 2 through 9 were retrieved from the cataloged run data group and were not part of the source input stream for this example. These statements correspond to statements with List Seq 3 through 10 in Figure 19-2.

```
-----------------------------------------------------------------------------------------------------------------------------------
  RUN     STMT    FILE    O N S D U T R A B S D S L  RB  R S M R S            START           END              SEQUENCE   LIST
  NAME    TYPE    NAME    L E E I P R E U U R L C S  PL  J O O F S            SEARCH          SEARCH           73 TO 80    SEQ
                          D W Q R D N P D F T M N T  TK  T P P O R
-----------------------------------------------------------------------------------------------------------------------------------
(USE_RDG ) (RC) (RDG1    ) (S)         (S)      (#,S)           (R)                                                          1
-----------------------------------------------------------------------------------------------------------------------------------
  RUN     STMT  LOGICAL  LBL  DTF/DD      KEY       KEY       KEY      QUALIFIER   USER I/O  ARRAY      PASSWORD/  SEQUENCE   LIST
  NAME    TYPE  FILENAME       NAME     FIELD-1   FIELD-2   FIELD-3   ICF CRD DBI   MODULE   QUAL       AUTH.-ID   73 TO 80    SEQ
-----------------------------------------------------------------------------------------------------------------------------------
(USE_RDG ) (RF) (CUSTOMER) (S) (M4OLD   )                                                                                        2
-----------------------------------------------------------------------------------------------------------------------------------
  RUN     STMT  CURRENT  DEL  DTF/DD     ALIAS   ALIAS  LOGICAL                                                   SEQUENCE   LIST
  NAME    TYPE   NAME    ETE   NAME      NAME    TYPE    UNIT                                                     73 TO 80    SEQ
-----------------------------------------------------------------------------------------------------------------------------------
(USE_RDG ) (LB) (ITMPRICE)    (M4OLD   ) (PRICE   ) (F)                                                                          3
-----------------------------------------------------------------------------------------------------------------------------------
  RUN     STMT  LOGICAL  LBL  DTF/DD      KEY       KEY       KEY      QUALIFIER   USER I/O  ARRAY      PASSWORD/  SEQUENCE   LIST
  NAME    TYPE  FILENAME       NAME     FIELD-1   FIELD-2   FIELD-3   ICF CRD DBI   MODULE   QUAL       AUTH.-ID   73 TO 80    SEQ
-----------------------------------------------------------------------------------------------------------------------------------
(USE_RDG ) (RF) (ITEM    ) (S) (M4CORD1 ) (ITEMORD )                    (O)                                                   4
-----------------------------------------------------------------------------------------------------------------------------------
  RUN     STMT  CURRENT  DEL  DTF/DD     ALIAS   ALIAS  LOGICAL                                                   SEQUENCE   LIST
  NAME    TYPE   NAME    ETE   NAME      NAME    TYPE    UNIT                                                     73 TO 80    SEQ
-----------------------------------------------------------------------------------------------------------------------------------
(USE_RDG ) (LB) (CUSTNO  )    (M4CORD1 ) (BOCUSTNO) (F)                                                                          5
(USE_RDG ) (LB) (ORDERNO )    (M4CORD1 ) (ITORDNO ) (F)                                                                          6
(USE_RDG ) (LB) (QTYBKORD)    (M4CORD1 ) (ITQTYBKO) (F)                                                                          7
-----------------------------------------------------------------------------------------------------------------------------------
  RUN     STMT  LOGICAL  LBL  DTF/DD      KEY       KEY       KEY      QUALIFIER   USER I/O  ARRAY      PASSWORD/  SEQUENCE   LIST
  NAME    TYPE  FILENAME       NAME     FIELD-1   FIELD-2   FIELD-3   ICF CRD DBI   MODULE   QUAL       AUTH.-ID   73 TO 80    SEQ
-----------------------------------------------------------------------------------------------------------------------------------
(USE_RDG ) (RF) (SUBFILE1) (S) (M4SUBF1 )                                                                                        8
-----------------------------------------------------------------------------------------------------------------------------------
  RUN     STMT  PARAMETER                                                                                          SEQUENCE   LIST
  NAME    TYPE    NAME                                PARAMETER VALUE                                              73 TO 80    SEQ
-----------------------------------------------------------------------------------------------------------------------------------
(USE_RDG ) (RP) (FREESIZE)        (100K                                                  )                                      9

                                                  ******************************
                                                  *  PROC NAME - MAIN          *
                                                  *  INPUT STREAM PROCEDURE    *
                                                  ******************************
-----------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                                 MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB      SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID           ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM      73 TO 80    SEQ
-----------------------------------------------------------------------------------------------------------------------------------
(ER)                                                                                (N)                                         10
     ***** ASL Procedure Follows - Using Translator Version x.xx
.
.
.
```

Figure 19-3 Example of Using a Run Data Group

# Procedures, Requests, and Request Groups

In addition to retrieving source objects stored in source libraries using the COPY command, you can also use the CATOLOG command to catalog procedures and requests to the common library. Once an element is cataloged, it can be listed, retrieved back into its source statement format, or included as input to a processing run directly from the library in lieu of providing the entire item instream.

If a procedure or request is valid, it is available for any cataloging action that you specify. (Cataloging procedures and requests is described in detail in the *VISION:Builder Reference Summary*.) An ASL procedure is cataloged in the same way as a fixed format request with the following changes:

- ##PROC and ##PEND are substituted for the ASL PROC and END PROC commands.
- The procedure is preceded by a fixed format ER statement.

A procedure or request can be cataloged as a separate item, as part of a request group, or both. This is performed in a VISION:Builder processing run by supplying the necessary CATALOG command or fixed format CT statements. The order of statement operations is important when performing maintenance on request groups. Refer to the discussion of Cataloging Operations in the Technical Notes chapter of the *VISION:Builder Reference Summary* for a discussion of these considerations.

Keep in mind that all elements of a request group are retrieved when a request group is selected for inclusion into an application and all elements of the group (procedures and requests) need to fit together in some coherent and logical fashion. For example, when a cataloged procedure references fields from a file with the appropriate qualifier, the application must use that file consistent with the designation of that qualifier.

A procedure or request can be listed from the library after being cataloged. This is performed in a processing run by supplying a CATALOG command or fixed format CT statement with the requested item names.

You can transfer any procedure or request back into its source statement format. The source statements are either in ASL format (for a procedure) or in the fixed format syntax (for a request). This transfer is performed during a source statement retrieval run.

In order to include a cataloged procedure, request, or request group in a processing run (instead of providing the source instream), you must provide a RETRIEVE command or fixed format CR statement for each procedure, request, or request group. VISION:Builder automatically retrieves the source for the item and includes it in the source stream for the application.

# Item Tracking

Item tracking allows you to monitor items stored in the common library by recording user data, expiration dates, retention periods, and date/time stamps for items stored in the library. You invoke item tracking by using the TRACK command or by placing IT statements in your definitions (files, tables, and so on). When an item is created, you can specify how long to save it in the library. At the same time, you can also schedule an item for automatic deletion. See the *VISION:Builder Reference Summary* for more information about the IT statement or the *ASL Reference Guide* for more information about the TRACK command.

With item tracking, your installation can optionally request that a date/time stamp be automatically recorded with each item cataloged in the common library. See the Installation and Support manual for information regarding the selection of this option. If this option is chosen, the data/time stamp is automatically maintained in the library and it reflects the date/time that the item is created, last updated, and last retrieved. This information cannot be externally specified. The date/time option is shown in the following outputs:

- Informational messages listed on the output listing when an item is retrieved.
- Library utility reports.
- Documentation generated from the library.
- File definition glossaries and lists.
- Listings produced by MARKUTIL list and dump operations.

## Processing with Multiple Common Libraries

Whenever the term *common library* has been used previously, the term always referred to a single entity that can be thought of as the logical common library. The logical common library used during an application run can consist of multiple

common library data sets logically concatenated in some order. Following are the rules and considerations that apply with regard to the use of multiple common libraries:

- The default file name (DTF/DD name) of the primary common library is M4LIB.
- The file names of additional (or alternate) common libraries in a run must be M4LIBn (where n is 1 through 9).
- The default order of search is M4LIB, M4LIB1, M4LIB2, ..., M4LIB9. This order can be overridden by using the MULTILIB command. The ORDER keyword is followed by ddnames that specify the order in which the common libraries will be searched.
- Multiple common libraries cannot be used in a definition/maintenance run. If the supplied JCL specifies more than one common library, only the primary library (M4LIB) is used.
- Multiple common library access is suppressed in an application run that can contain JCL for multiple common libraries whenever any of the following conditions are true:
  - A CATALOG command or fixed format CT statement is present.
  - A transaction file❹ is specified but no explicit transaction group is specified. This occurs when no GROUPS keyword is specified on the FILE TRAN command or no fixed format RG statement is included.
  - An ASL MULTILIB OFF command or a fixed format RP MULTILIB OFF statement is present.
  - A single ddname is specified on the MULTILIB command.
- When searching multiple common libraries, request groups are located before requests, and run data groups are located before field definitions. Take this into consideration when trying to access requests/request groups, or logical data views and file definitions with the same name.

See the *VISION:Builder ASL Reference Guide* for more information about the MULTILIB command.

See the *VISION:Builder Reference Summary* for more information about the RP MULTILIB statement.

Chapter

# 20 Program Analyzer

The ability to debug and document a program is often as important as the coding of that program. For the development and implementation of a program, you need the ability to evaluate and investigate any errors or problems that can occur.

For the maintenance of a program, documenting the intricacies of processing and field usage becomes essential. The information you need to perform these duties ranges from a cross reference of all fields accessed in the program to an execution trace of the processing performed by the program. The VISION:Builder program analyzer (PAL) allows you to create reports that meet these needs.

Producing the reports is a two-step process. First, while executing your application, VISION:Builder generates a file (M4PAOUT) containing all the requested records. Once this file is generated, you, in turn, process the file with VISION:Builder supplied file definitions and requests producing the program analyzer reports.

PAL can also be used to help streamline a program's processing to make best use of your resources. PAL produces messages that inform you of the automatic numeric conversions performed by VISION:Builder. While the conversions save you much time and effort, they do consume computing resources. These messages allow you to avoid unnecessary automatic conversions, thus making best use of your resources.

Note that the information produced by PAL is expressed in terms of the fixed format statements in VISION:Builder. If you are using ASL, add the LISTGEN keyword to the CONTROL command in your application, which causes VISION:Builder to list the underlying fixed format statements generated for the ASL procedures in your application. Otherwise, it can be very difficult to relate the PAL output back to the original source statements of your application.

The program analyzer capability is available in VISION:Builder through either ASL or the fixed format syntax and in the VISION:Workbench for ISPF. It is not available in VISION:Workbench for DOS.

# Numeric Conversion Messages

Chapter 4, *Field Characteristics and Handling* describes how VISION:Builder automatically converts different field types to a common field type when performing relational and arithmetic operations. These automatic conversions require processing resources. If there are a large number of conversions, processing efficiency can be reduced.

PAL enhances your source statement listings with messages indicating where, what type, and how many automatic conversions were performed for an operation. You can eliminate any unnecessary conversions that are occurring. Numeric conversion messages are generated when you include the DOCUMENT command with the CONVMSGS keyword in the Run Control command group. These numeric conversion messages appear directly on the source listing. They are not written out to the program analyzer file (M4PAOUT).

Figure 20-1 illustrates the types of conversions that VISION:Builder performs and the steps taken in those conversions. The table on page 20-2 illustrates the conversions that take place when an arithmetic or logical operation occurs between multiple operands on any processing statement. The table on page 20-3 illustrates the conversions that take place when a value is placed into a resulting operand.

**Note:** The boxes detailing conversion steps in this table list the steps for each operand, A on the first line, and B on the second line.

This table shows the automatic numeric conversion involving operands A and B.

| Operand A Field Type | Operand B Field Type: C | Z | P | F | E | Notes |
|---|---|---|---|---|---|---|
| **C** | C-Z-P<br>C-Z-P[1] | C-Z-P<br>Z-P | C-Z-P<br>- | C-Z-P<br>F-P | C-Z-P-F-E<br>- | [1] Converted only for arithmetic operations. |
| **Z** | Z-P<br>C-Z-P | Z-P<br>Z-P | Z-P<br>- | Z-P<br>F-P | Z-P-F-E<br>- | |
| **P** | -<br>C-Z-P | -<br>Z-P | -<br>- | -<br>F-P | P-F-E<br>- | |
| **F** | F-P<br>C-Z-P | F-P<br>Z-P | F-P<br>- | F-P<br>F-P[2] | F-E<br>- | [2] Converted only if differing scales are involved. |
| **E** | -<br>C-Z-P-F-E | -<br>Z-P-F-E | -<br>P-F-E | -<br>F-E | -<br>- | |

*C* - Character *Z* - Zoned *P*- Packed *F*- Fixed *E*- Floating Point

This table shows the automatic numeric conversion involving operand C.

| | | Operand C Field Type | | | | |
|---|---|---|---|---|---|---|
| | | **C** | **Z** | **P** | **F** | **E** |
| **Sending Field Type** | **C** | - | C-Z | C-Z-P | C-Z-P-F | C-Z-P-F-E |
| | **Z** | Z-C | - | Z-P | Z-P-F | Z-P-F-E |
| | **P** | P-Z-C | P-Z | - | P-F | P-F-E |
| | **F** | F-P-Z-C | F-P-Z | F-P | - | F-E |
| | **E** | E-F-P-Z-C | E-F-P-Z | E-F-P | E-F | - |

*C* - Character *Z* - Zoned *P*- Packed *F*- Fixed *E*- Floating Point

VISION:Builder handles the conversions automatically. For example, when two character fields are used for arithmetic, two conversions per operand are performed. They are first converted to zoned decimal and then converted to packed. When moving values into resulting fields (operand C), VISION:Builder ensures that the field type of the sending field is the same as the receiving field.

In Figure 20-1, the temporary fields are defined as zoned fields (field type Z). If you look at the conversion messages, you see that VISION:Builder changes the zoned fields into packed fields to perform the arithmetic and converts the fields into zoned to place that field into another temporary field. If you define the temporary fields as packed (field type P), you would reduce the number of conversions. Figure 20-2 illustrates the listing of the new run with the reduced number of conversions.

```
INVOICE:  FIELD LEN 4   TYPE Z   DEC 2
RATE:     FIELD LEN 3   TYPE Z   DEC 2  INIT .07
SALESTAX: FIELD LEN 4   TYPE Z   DEC 2
TOTAL:    FIELD LEN 6   TYPE Z   DEC 2
TOTPRICE: FIELD LEN 6   TYPE Z   DEC 2
TOTFRT:   FIELD LEN 3   TYPE Z

PALEX1:  PROC TYPE TYPE3 ;PAL examples of uneccesary conversions
  LET T.TOTAL    = 0
  LET T.INVOICE  = T.INVOICE + 1
  LET T.TOTPRICE =   ITMPRICE * X.QTYSHIP
  LET T.TOTFRT   = X.FRTCOST * X.QTYSHIP
  LET T.SALESTAX = T.RATE * T.TOTPRICE
  LET T.TOTAL    = T.TOTPRICE + T.SALESTAX + T.TOTFRT
END PROC

-------------------------------------------------------------------------------------------------------------------------------
STMT   FIELD    FIELD FLD DEC OUTPT EDIT      INITIAL         LINE 1 OF       LINE 2 OF                              SEQUENCE   LIST
TYPE   NAME     LNGTH TYP PLC EDIT  LGTH       VALUE        COLUMN HEADING  COLUMN HEADING                          73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(TF) (TEMP__11) (  7) (P) (2)                                                                                      (ASLGEN'D)    20
-------------------------------------------------------------------------------------------------------------------------------
STMT SEQ  LOG CON .....OPERAND-A.....  OPER  .............OPERAND-B.............   ......RESULT......  PARTIAL FIELD  SEQUENCE   LIST
TYPE NO.  LEV CTR QLF FIELD   SEG-LVL  AIN   QLF FIELD   SEG-LVL  (OR CONSTANT)   QLF FIELD  SEG-LVL  STRT NUMB OPR  73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(PR) (100)                             (R )   (D,0                          ) (T,TOTAL    )                        (ASLGEN'D)    21
** MK4Z201  TYPE 0                                                             P-Z          CONVERSIONS -  1 .                 *****
(PR) (101)          (T,INVOICE )       (+ )   (D,1                          ) (T,INVOICE )                        (ASLGEN'D)    22
** MK4Z201  TYPE 0   Z-P                        Z-P                            P-Z          CONVERSIONS -  3 .                 *****
(PR) (102)          (ITMPRICE) 005-3   (* )   (X,QTYSHIP ) 001-1              (T,TOTPRICE)                          (ASLGEN'D)    23
```

Figure 20-1 Unnecessary Numeric Conversions (Page 1 of 2)

```
** MK4Z201  TYPE 0   Z-P                          Z-P                                  P-Z          CONVERSIONS -  3 .            *****
(PR) (103)         (X,FRTCOST ) 001-1  (* )   (X,QTYSHIP ) 001-1                  (T,TOTFRT  )                        (ASLGEN'D)    24
** MK4Z201  TYPE 0   Z-P                          Z-P                                  P-Z          CONVERSIONS -  3 .            *****
(PR) (104)         (T,RATE    )        (* )   (T,TOTPRICE)                         (T,SALESTAX)                        (ASLGEN'D)    25
** MK4Z201  TYPE 0   Z-P                          Z-P                                  P-Z          CONVERSIONS -  3 .            *****
(PR) (105)         (T,TOTPRICE)        (+ )   (T,SALESTAX)                         (T,TEMP__11)                        (ASLGEN'D)    26
** MK4Z201  TYPE 0   Z-P                          Z-P                                               CONVERSIONS -  2 .            *****
(PR) (106)         (T,TEMP__11)        (+ )   (T,TOTFRT  )                         (T,TOTAL   )                        (ASLGEN'D)    27
** MK4Z201  TYPE 0                                Z-P                                  P-Z          CONVERSIONS -  2 .            *****
```

Figure 20-1 Unnecessary Numeric Conversions (Page 2 of 2)

```
PINVOICE:  FIELD LEN 4   TYPE P   DEC 2
PRATE:     FIELD LEN 3   TYPE P   DEC 2  INIT .07
PSALESTX:  FIELD LEN 4   TYPE P   DEC 2
PTOTAL:    FIELD LEN 6   TYPE P   DEC 2
PTOTPRIC:  FIELD LEN 6   TYPE P   DEC 2
PTOTFRT:   FIELD LEN 3   TYPE P
END PROC

PALEX2:  PROC TYPE TYPE3 ;PAL examples without unnecesary conversions
  LET T.PTOTAL    = 0
  LET T.PINVOICE  = T.PINVOICE + 1
  LET T.PTOTPRIC  =   ITMPRICE * X.QTYSHIP
  LET T.PTOTFRT   = X.FRTCOST * X.QTYSHIP
  LET T.PSALESTX  = T.PRATE * T.PTOTPRIC
  LET T.PTOTAL    = T.PTOTPRIC + T.PSALESTX + T.PTOTFRT
END PROC

-------------------------------------------------------------------------------------------------------------------------------------
STMT   FIELD    FIELD FLD DEC OUTPT EDIT      INITIAL          LINE 1 OF        LINE 2 OF                          SEQUENCE   LIST
TYPE   NAME     LNGTH TYP PLC EDIT  LGTH       VALUE         COLUMN HEADING   COLUMN HEADING                       73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------------
(TF) (TEMP__23) (  7) (P) (2)                                                                                       (ASLGEN'D)    29
-------------------------------------------------------------------------------------------------------------------------------------
STMT SEQ  LOG CON .....OPERAND-A.....  OPER  ............OPERAND-B.............   ......RESULT......  PARTIAL FIELD  SEQUENCE   LIST
TYPE NO.  LEV CTR QLF FIELD   SEG-LVL  ATN   QLF FIELD   SEG-LVL  (OR CONSTANT)   QLF FIELD  SEG-LVL  STRT NUMB OPR  73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------------
(PR) (100)                            (R )   (D,0                              ) (T,PTOTAL  )                        (ASLGEN'D)    30
(PR) (101)         (T,PINVOICE)       (+ )   (D,1                              ) (T,PINVOICE)                        (ASLGEN'D)    31
(PR) (102)           (ITMPRICE) 005-3 (* )   (X,QTYSHIP ) 001-1                  (T,PTOTPRIC)                        (ASLGEN'D)    32
** MK4Z201  TYPE 0   Z-P                          Z-P                                               CONVERSIONS -  2 .            *****
(PR) (103)         (X,FRTCOST ) 001-1  (* )   (X,QTYSHIP ) 001-1                  (T,PTOTFRT )                        (ASLGEN'D)    33
** MK4Z201  TYPE 0   Z-P                          Z-P                                               CONVERSIONS -  2 .            *****
(PR) (104)         (T,PRATE   )        (* )   (T,PTOTPRIC)                         (T,PSALESTX)                        (ASLGEN'D)    34
(PR) (105)         (T,PTOTPRIC)        (+ )   (T,PSALESTX)                         (T,TEMP__23)                        (ASLGEN'D)    35
(PR) (106)         (T,TEMP__23)        (+ )   (T,PTOTFRT )                         (T,PTOTAL  )                        (ASLGEN'D)    36
```

Figure 20-2 Efficient Use of Automatic Conversions

Using the conversion messages in this manner, you can avoid unnecessary processing that can reduce the efficiency of your operation.

# Generating the M4PAOUT File

The program analyzer generates a file (M4PAOUT) of cross reference and execution trace data for an application. The file is generated by including a DOCUMENT command in the application and a DTF/DD statement for M4PAOUT in the JCL. XREF and EXEC TRACE keywords on the DOCUMENT command determine if cross reference or execution trace records or both are written to the file. The generated numeric conversion messages are not written to M4PAOUT. However, the field cross reference listing identifies the references to the field that involve conversions and lists the total number of conversions for all references to the field.

Once the M4PAOUT file is created, you generate the various trace and cross reference reports through a VISION:Builder processing run using the supplied file definitions and requests. A user coordinated file specifies which reports you want, while the requests select and format the data in the proper manner. Source that Generates Reports on page 20-24 describes the generation of these reports in more detail.

Note that an execution trace record is written to the M4PAOUT file every time control passes to or from a procedure, report, subfile, or request. You may want to limit the number of records written to the file. You can set the maximum number of lines you want generated for the trace and limit the size of the trace report by specifying MAXLINES on the DOCUMENT command.

You can also control the block size of the M4PAOUT file through the JCL. Note that the file can only be a variable length sequential data set.

If the operating system supports system-determined block size and the M4PAOUT file meets the qualifications for this capability, the file is created with the optimum block size determined by the operating system (see Chapter 7, *Output File Processing* for more information about system-determined block size support).

Figure 20-3 illustrates a DOCUMENT command that requests both cross reference and execution trace data and specifies a maximum number of lines for the trace, as well as the conversion messages.

```
DOCUMENT CONVMSGS, XREF, EXECTRACE, MAXLINES 2000
```

Figure 20-3 DOCUMENT Command Requesting Cross Reference and Execution Trace Data

Figure 20-4 illustrates the flow of an application run's processing step that produces the M4PAOUT file.

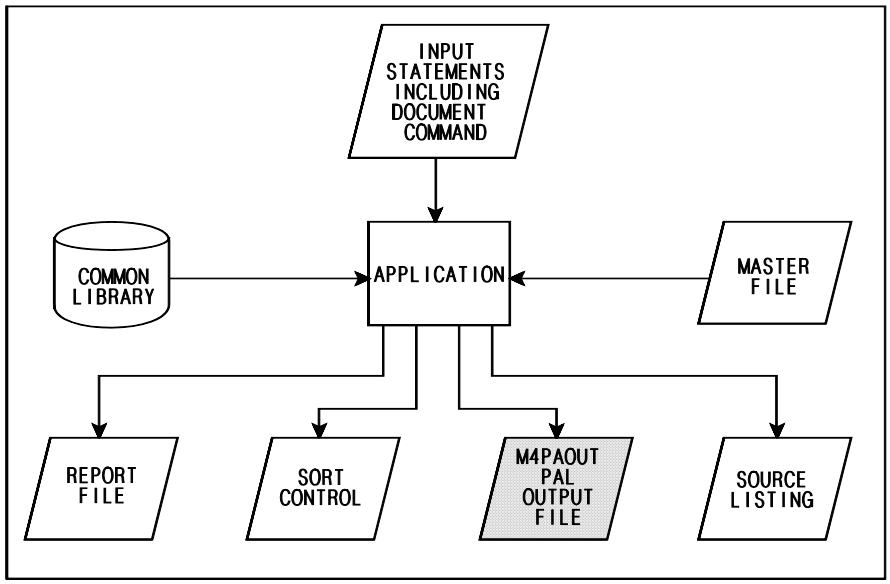



Figure 20-4 Processing Flow to Produce M4PAOUT

Note that the VISION:Builder input statements include a DOCUMENT command that requests both cross reference and trace reports. If either report is requested, the M4PAOUT output file is produced.

# Processing the M4PAOUT File

Computer Associates provides the file definitions and requests with VISION:Builder so that you can process the M4PAOUT file with VISION:Builder. Source that Generates Reports on page 20-24 discusses the provided file definitions and requests to aid you in creating your own reports. If you select trace information only, you do not need to sort M4PAOUT before processing it. The JCL, including sort control statements, needed to sort and process the M4PAOUT file and produce the PAL reports is provided in the *VISION:Builder for OS/390 Environment Guide* (OS/390 users) or the *VISION:Builder for VSE Environment Guide* (VSE users).

To produce the provided reports, you need to input a selection record to the run as coordinated file number one (M4CORD1). With this record, you can select any or all of the PAL reports. Figure 20-5 presents the VISION:Builder glossary of the file definition for the selection record.

```
                                        DETAILED GLOSSARY BY LOCATION FOR
                                           FILE DEFINITION - IGCPALRS                                    PAGE 1
FILE IDENTIFICATION =                                                          RECORD FORMAT = FIXED UNBLOCKED
NUMBER OF SEGMENTS IN FILE = 1                                                 RECORD SIZE = 80
NUMBER OF FIELDS IN FILE = 8                                                   BLOCK SIZE = 80

                                         ******************************
                                         *     SEGMENT  1, LEVEL 1    *
                                         ******************************
SEGMENT OCCURS N TIMES = 1                                          KEY FIELD 1 = REP-SEL    TYPE = C   LENGTH = 7
SEGMENT SIZE = 16
NUMBER OF FIELDS IN SEGMENT = 8
-----------------------------------------------------------------------------------------------------------------------
FIELD   FIELD   FIELD    FIELD   FIELD    DEC    CNT FIELD  EDIT CODES    EDIT    OUTPUT   LINE          COLUMN           SEARCH
NAME    TYPE   LOCATION  LENGTH  RNDING  PLACES  FOR SGMT   ( ) ( ) ( )  LENGTH   WIDTH     NO   ***    HEADING    ***   ALIAS
-----------------------------------------------------------------------------------------------------------------------
REP-SEL   C        1       7                                              7       7                                    2
SELECTS   C       11       6                                              6       6                                    2
XREFNAME  C       11       1                                              1       1                                    2
XREFC/P   C       12       1                                              1       1                                    2
XREFBRAN  C       13       1                                              1       1                                    2
XREFFILE  C       14       1                                              1       1                                    2
TRAC-REG  C       15       1                                              1       1                                    2
TRAC-MAP  C       16       1                                              1       1                                    2
```

Figure 20-5 Record Selection File Glossary

- Y in position 11 selects the field name cross reference report.
- Y in position 12 selects the constant/pattern cross reference report.
- Y in position 13 selects the branch location cross reference report.
- Y in position 14 selects the file cross reference report.
- Y in position 15 selects the request execution trace report.
- Y in position 16 selects the mapping request execution trace report.

This selection record is processed as a user coordinated file along with the M4PAOUT file processed as the master file when you produce the PAL reports. The reports produced by the supplied requests are each described in detail in the following sections. Note that you can tailor these requests to best meet your installation's needs.

Figure 20-6 shows the flow of a program analyzer processing run, using the provided requests.

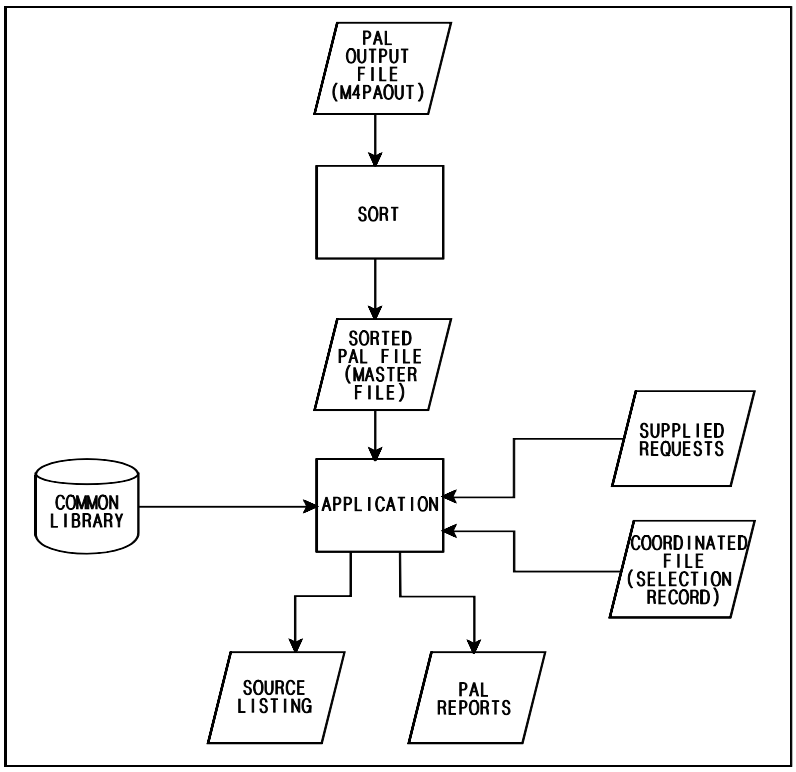



Figure 20-6 Processing of M4PAOUT to Produce PAL Reports

# Cross Reference Reports

You can generate four different cross reference reports from M4PAOUT. Each report lists information about a particular type of referenced item. The items are field names, constants/patterns, files, and branch locations. Each report defines an item and defines its references within the application. All references are made with respect to the fixed format syntax form of the application. In addition, the source listing should be produced so that all ASL coded procedures also show the generated fixed format syntax statements. This is done by including the LISTGEN keyword on the CONTORL command.

## Field Name Cross Reference Report

The field name cross reference report is a listing of the fields referenced by the procedures, requests, reports, and subfiles in your application. Arrays, tables, and segments referenced are also included. The report includes a description of the field and an explanation of the reference. The field description includes:

- Field qualifier
- Field name
- Field type
- Segment number
- Level number

The reference explanation includes:

- Request name (the request in which the reference took place).
- Request type.
- Source listing sequence number (where in the application the reference took place).
- Reference type.

Note that the source listing sequence number, also called the list sequence number, is the number that prints on the far right hand side of the source listing and corresponds to each line of source code input by the VISION:Builder user.

Figure 20-7 illustrates an example of the field name cross reference report.

```
JAN 15, 1995                    FIELD NAME CROSS REFERENCE FOR UPDTSEG , 01/15/95, 18.34.10                    PAGE  1
-----------------------------------------------------------------------------------------------------------------------------
.....FIELD..............  .....REFERENCES........
QLF  NAME     TYPE SEG-LVL  REQUEST  TYPE  LIST-SEQ.....(AND REFERENCE TYPE)                                    TOTAL CNV.
-----------------------------------------------------------------------------------------------------------------------------
    BKORDQTY  Z     2-2    UPDATE    3    *00018C   000033R1                                                          1
    BKORDSEG  SG    1-1    UPDATE    3    000017A

    CUSTNO    CK    2-2    UPDATE    3    000019C   000035R1

    ITEMNAME  C     1-1    UPDATE2   3    000043R1
    ITEMNO    CK    1-1    UPDATE    3    000022R1
    ITEMNO    CK    1-1    UPDATE2   3    000042R1
    ORDERNO   CK    2-2    UPDATE    3    000020C   000037R1
    ORDQTY    Z     1-1    UPDATE    3    *00014A   *00014C   000029R1                                              2
    QANONHND  Z     1-1    UPDATE    3    *00010C   *00014B   000025R1                                              2
    QTYBKORD  Z     1-1    UPDATE    3    *00015B   *00016C   000031R1                                              2
    SHIPYTD   Z     1-1    UPDATE    3    *00015A   *00015C   000027R1                                              2
 F  DELETE    F            UPDATE    3    *00021C                                                                   1
 O  BKORDQTY  Z     2-2    UPDATE    3    000032R1
 O  BKORDQTY  Z     2-2    UPDATE2   3    000044R1
 O  CUSTNO    CK    2-2    UPDATE    3    000034R1
 O  CUSTNO    CK    2-2    UPDATE2   3    000045R1
 O  ORDERNO   CK    2-2    UPDATE    3    000036R1
 O  ORDERNO   CK    2-2    UPDATE2   3    000046R1
 O  ORDQTY    Z     1-1    UPDATE    3    000028R1
 O  QANONHND  Z     1-1    UPDATE    3    000024R1
 O  QTYBKORD  Z     1-1    UPDATE    3    000030R1
 O  SHIPYTD   Z     1-1    UPDATE    3    000026R1
 T  FILR      C            UPDATE    3    000007E
 T  FILR      C            UPDATE2   3    000047R1
 X  CODE      C            UPDATE    3    000008A   000012A   000023R1
 X  CODE      C            UPDATE2   3    000040A
```

Figure 20-7 Field Name Cross Reference Report

Each column on the report contains specific information that is described below.

FIELD QLF — The implicit or explicit qualifier for this field, array, table, or segment.

| | |
|---|---|
| blank or N | Field from new master work area. |
| O or 0 | Field from old master work area. |
| 1-9 | Field from coordinated file 1-9. |
| F | Flag name. |
| S | STEMP field (available in formatted sectional and formatted reporting). |
| T | Temporary field name. |
| V | Linkage section. |
| W | Working storage. |
| X | Transaction file. |
| A, B, E, G, H, I, K, M, Q | Array. |

FIELD NAME — The name of the referenced field.

FIELD TYPE — The field type of the referenced field as coded in the application or in a definition stored in the common library. In addition to the standard field types, the following are used:

| Code | Identifies |
|---|---|
| CK | Character key field. |
| CV | Variable length field. |
| xR | Automatic table lookup result field, where x is a field type. |
| xK | Record or segment key, where x is a field type. |
| SG | Segment name. |
| AD | Array name. |
| TB | Table name. |
| RK | Record key referenced by an RD, RE, or RG operator. |
| U | Unidentified or invalid field name. |

| | |
|---|---|
| FIELD SEG-LVL | The segment and level numbers of the segment in which the referenced field is contained as specified in a file definition. Virtual segments are always level V. |
| REFERENCE REQUEST | The name of the request that references the field. Note that the ASL translator may, in some cases, generate additional subroutine procedures from a procedure, each with a name determined by the ASL translator. These generated names are different from the procedure name and appear as the name of the procedure that references the field. |
| REFERENCE TYPE | The type of request that references the field. In addition to the codes normally used for request type, GS is used to identify a GDBI (mapping) subroutine. GDBI is described in the *GDBI Programming Manual*. Contact Technical Support if you do not have a copy of this book. |
| REFERENCE LIST-SEQ with REFERENCE TYPE | The source listing sequence number of the fixed format request, control, or output statement that references the field, along with the specific type of reference. An asterisk (*) in front of the list sequence number indicates that an automatic conversion is required at this reference. |

| Code | Identifies Type of Reference |
|---|---|
| A,B,C | Operand of the PR statement. A second character of P indicates the field is partial fielded, while a second character of S (START) or N (NUMBER) indicates partial fielding with a flag. |
| Fn | Formatted or formatted sectional output specification. |
| G | GSI call parameter. |
| R | RF statement specification (for master file or transaction❹ only). |
| Rn | Output specification. |
| RD,RE, RG | Used by these operators on a PR statement. |
| I,E | Implicit or explicit definition of a temporary field. |

| | |
|---|---|
| TOTAL CNV | The number of automatic conversions performed on a field in a request. Note that the number of conversions is calculated separately by request and is indicated by an asterisk (*) in front of the list sequence numbers. |

Look at the boxed item in Figure 20-7 and note that the field name is CUSTNO. This field is from the new master file work area as indicated by the blank under QLF. The CK indicates that the field is character and a key field. It is positioned in the second level and second segment of the file as represented by the 2-2. This reference took place in a request named UPDATE, which is a type 3 request.❹ Two references actually occurred in this request: one in the C operand of line 19 and another in the R1 statement of line 35. All references to CUSTNO in request UPDATE are listed on the same line of the report. Note that all references to a field within a single request list on the same line of the report. The line folds, if necessary, to print all the references.

## Constant/Pattern Cross Reference Report

The constant/pattern cross reference report is a listing of the constants and patterns used by the procedures and requests in your application. The report includes a description of the constant or pattern and an explanation of the reference. The constant or pattern description includes:

- Constant or pattern qualifier.
- Constant or pattern value.

The reference explanation includes:

- Request name (the request in which the constant or pattern appeared).
- Request type.
- Source listing sequence number (where in the application the reference took place).
- Reference type.

Figure 20-8 illustrates the constant/pattern cross reference report.

```
JAN 15, 1995                CONSTANT/PATTERN CROSS REFERENCE FOR UPDTSEG , 01/15/95, 18.34.10                PAGE  1
-------------------------------------------------------------------------------------------------------------------------
.....FIELD.....................  ..... REFERENCES.....
QLF  NAME                       REQUEST  TYPE  LIST-SEQ.....(AND REFERENCE TYPE)                                TOTAL CNV.
-------------------------------------------------------------------------------------------------------------------------
 C                              UPDATE    3    000018  000019  000020
 C  I6                          UPDATE    3    000008
 C  I7                          UPDATE    3    000012

 C  I7                          UPDATE2   3    000040


 D  0                           UPDATE    3    000010  000016

 D  2                           UPDATE    3    000021
```

Figure 20-8 Constant/Pattern Cross Reference Report

Information about each referenced constant and pattern appears as follows on the constant/pattern cross reference report:

| | |
|---|---|
| CONSTANT/ PATTERN QLF | The qualifier for a constant or pattern. |

| | |
|---|---|
| CONSTANT/ PATTERN NAME | The value of a constant or pattern. |
| REFERENCE REQUEST | The name of a request that uses a constant or pattern. |
| REFERENCE TYPE | The type of request that uses a constant or pattern. In addition to the codes normally used for request type, GS is used to identify a GDBI (mapping) subroutine. |
| REFERENCE LIST-SEQ with REFERENCE TYPE | The source listing sequence number of the fixed format request, control, or output statement that references the field, along with the specific type of reference: |

| Code | Identifies Type of Reference |
|---|---|
| A,B,C | Operand of the PR statement. A second character of P indicates the field is partial fielded, while a second character of S (START) or N (NUMBER) indicates partial fielding with a flag. |
| Fn | Formatted or formatted sectional output specification. |
| G | GSI call parameter. |
| R | RF statement specification (for master file or transaction❹ only). |
| Rn | Output specification. |
| RD,RE, RG | Used by these operators on a PR statement. |
| I,E | Implicit or explicit definition of a temporary field. |

| | |
|---|---|
| TOTAL CNV | The number of automatic conversions performed on a constant in a request. Note that the number of conversions is calculated separately by request. |

Look at the boxed items in the report and note that there are two of the cross-referenced constants. The character constant is I7; it is referenced in the type 3 request❹ named UPDATE2 in line 40 of the VISION:Builder source. The decimal constant is zero and is referenced in the type 3 request❹ named UPDATE in lines 10 and 16 of the VISION:Builder source.

## Branch Location Cross Reference Report

The branch location cross reference report is a listing of the branch locations used by each request in your application. The report includes a description of the location to which you branch and a description of the location from which you branched.

The description of the branched-to location includes:

- Name of request branched to or name of VISION:Builder maintained location (END or OUTPUT).
- Type of request.
- Source listing sequence number to which you branch.

The description of the branched-from location includes:

- Name of request from which you branched.
- Type of request.
- Source listing sequence number from which you branched.

Figure 20-9 illustrates a branch location cross reference report.

```
JAN 15, 1995                BRANCH LOCATION CROSS REFERENCE FOR UPDTSEG , 01/15/95, 18.34.10                PAGE  1
-------------------------------------------------------------------------------------------------------------------------
......BRANCH LOCATION......   .....FROM REFERENCE.....
 REQUEST     TYPE  LIST-SEQ   REQUEST  TYPE  LIST-SEQ.....
-------------------------------------------------------------------------------------------------------------------------
END                          UPDATE    3    00013
OUTPUT                       UPDATE    3    00011
UPDATE        3    00012     UPDATE    3    00009
```

Figure 20-9 Branch Location Cross Reference Report

Information about each branch location used in your application appears on the branch location cross reference report as follows:

| | |
|---|---|
| BRANCH LOCATION REQUEST | The name of a branch-to location. This is either a request name or a VISION:Builder maintained location (for example, END). |
| BRANCH LOCATION TYPE | The request type if a branch-to location is in a request. This field is blank if the branch-to location is a VISION:Builder maintained location (for example, END). |
| BRANCH LOCATION LIST-SEQ | The source listing sequence number of a branch-to location within a request. This field is not used for branching beyond the branching request. |
| FROM REFERENCE REQUEST | The name of a request from which a branch location is referenced. |

| | |
|---|---|
| FROM REFERENCE TYPE | The type of request from which a branch location is referenced. |
| FROM REFERENCE LIST-SEQ | The source listing sequence number from which a branch location is referenced. This field is listed for all branches. |

If you look at the branch location report, you see that three different locations are referenced. Two are VISION:Builder locations that correspond to portions of the UPDATE request: END and OUTPUT. These are accessed from lines 13 and 11 of the UPDATE request, respectively. The other branch location is line 12 of the UPDATE request itself and it is accessed from line 9 of the same request.

## File Cross Reference Report

The file cross reference report is a listing of all the user files activated in a run. The report includes a description of a number of file specifications and a description of the specifications made when output files are created. The file specifications include:

- File name
- Logical unit assignment for VSE
- Array qualifier (if an array is defined)
- GDBI mode
- DTF/DD name
- File or array definition name
- DBD name
- Name of user I/O module

The output file specifications are listed under En-File and include:

- File name
- Request name
- Source listing sequence number
- Qualifier for entire record selection
- Request type
- En statement number (E1, E2, and so on)

Figure 20-10 is a file cross reference report.

```
JAN 15, 1995                          FILE REFERENCE FOR INVOICE , 01/15/95, 14.14.47                          PAGE  1
-------------------------------------------------------------------------------------------------------------------
FILE       DTF/DD  DOS   FD/AD    ARR   DBD     GDBI  USER I/O  ...EN-FILE.....(REFERENCES).....
NAME        NAME   ASSGN  NAME    QLF   NAME    MODE   MODULE     NAME   REC  REQUEST  TYP  LIST-SEQ.....
-------------------------------------------------------------------------------------------------------------------
M4OLD      *SAME*  *NONE* CUSTOMER
M4NEW      *SAME*  *NONE* CUSTOMER
M4TRAN     *SAME*  *NONE*
M4CORD1    *SAME*  *NONE* INVNUM
M4SUBF1    *SAME*  *NONE*                                        INVNUM     EOF       E   00105E1
M4SUBF2    *SAME*  *NONE*                                        INVINFO    PROC      M   00087E2
M4REJCT    *SAME*  *NONE*
SUMRY1     *SAME*  SYSO56                                        SUMRY1     PROCA     N   00100E1
```

Figure 20-10 File Cross Reference Report

Information about the files referenced in a run appear on the file cross reference report as follows:

| | |
|---|---|
| FILE NAME | The VISION:Builder default file name. For Report Summary Files, this is the DTF/DD or file name entered on the En statement. For Alternate List Files, this is the literal M4LIST1. |
| DTF/DD NAME | The VSE DTF name or OS/390 DD name for the file as specified in an OVERRIDE command or fixed format RP OVERRIDE statement, or an En statement, or the literal *SAME* if the name is the same as the file name. |
| DOS ASSGN | The logical unit assignment (SYSnnn) used in VSE. If overridden on an OVERRIDE command or fixed format RP OVERRIDE statement, the specified value prints instead of the default logical unit assignment assigned by VISION:Builder. The literal *NONE* prints if the system is not VSE. |
| FD/AD NAME | The name of a file or array definition in the VISION:Builder library to be associated with this file for this run. |
| ARR QLF | The qualifier used with this file, if it is an array. |
| DBD NAME | The IMS database name entered on an FD statement for this file, if it is an IMS database. |
| GDBI MODE | The associated setting of the MODE flag if the file is a GDBI (mapped) file. |
| USER I/O MODULE | The name of a user I/O module specified for this file on a FILE command or fixed format RF statement. |
| EN-FILE NAME | The specification made for DTF/DD name on an EXTRACT command, REPORT command, or En statement. |
| EN-FILE REC | The qualifier of the appropriate file or array if the file is a subfile with entire record selection specified. |
| REFERENCE REQUEST | Either the name of a request in which a subfile, alternate report, or alternate list file is referenced or the literal *RUNGRP* for references to any user file. |
| REFERENCE TYPE | Type of request in which a subfile, alternate report, or alternate list file is referenced. |

| | |
|---|---|
| REFERENCE LIST-SEQ with REFERENCE TYPE | The source listing sequence number of the request statement for this reference, along with the specific type of reference: |

| Code | Identifies |
|---|---|
| En | Output specification |
| RF | Additional file specification |
| RP | Run parameter |

Look at the file cross reference report and note that each file used in the application is listed on this report. Both the old and new master files have the CUSTOMER file definition associated with them. Two subfiles are output by this application. One is named INVNUM and is output by a type E request named EOF; the reference takes place on line 105 of the application in an E1 statement. The other subfile is named INVINFO and is output by a type M request named PROC; the reference takes place on line 87 of the application in an E2 statement.

# Request Execution Trace Reports

The program analyzer produces two different execution trace reports. Both trace the execution of the requests in your application.

- The request execution trace report analyzes the processing of all the requests and lists the settings of various flags.
- The mapping request execution trace report analyzes the processing of all mapping requests within the application.

## The Request Execution Trace Report

The request execution trace report shows the flow of request execution in your application. Because the processing flow is determined by the master file and transaction file,❹ the record number being accessed is also listed. This report also shows when each request is accessed and on which record the processing took place. The report also lists the values of various fields and the settings of several internal VISION:Builder flags at each request's beginning and ending point.

Figure 20-11 illustrates a request execution trace report.

```
JAN 15, 1995                    REQUEST EXECUTION TRACE FOR CUSTICF , 01/15/95, 19.27.18                    PAGE  1
-----------------------------------------------------------------------------------------------------------------------------
REF-SEQ B/E OUT REQUEST  TYPE REQ  MASTER   M4TRAN  ASTATUS DELETE   ECORD   MPASS     EOF     LSTAT RETURN   STRAN   TRAN XTRAN
                                                                   123456789       OT123456789              123456789
-----------------------------------------------------------------------------------------------------------------------------
    1   B      TEST      2    1        5        1          0    XXXXXXXXX       NNNNNNNNNNN
    2   E   Y  TEST      2    1        5        1          0    MXXXXXXXX       NNNNNNNNNNN
    3   B      REJ       4    1        5        1          0    MXXXXXXXX       NNNNNNNNNNN              M          4   11
    4   E   Y  REJ       4    1        5        1          0    MXXXXXXXX       NNNNNNNNNNN              M          4   11
    5   B      TEST      2    1        8        2          0    MXXXXXXXX       NNNNNNNNNNN
```

Figure 20-11 Request Execution Trace Report (Page 1 of 2)

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 6 | E | Y | TEST | 2 | 1 | 8 | 2 | 0 | MXXXXXXX | NNNNNNNNNN | | | |
| 7 | B | | SHIP | 3 | 1 | 8 | 2 | 0 | MXXXXXXX | NNNNNNNNNN | MMI | 1 | |
| 8 | E | Y | SHIP | 3 | 1 | 8 | 2 | 0 | MXXXXXXX | NNNNNNNNNN | MMI | 1 | |
| 9 | B | | TEST | 2 | 1 | 8 | 3 | 0 | MXXXXXXX | NNNNNNNNNN | | 1 | |
| 10 | E | | TEST | 2 | 1 | 8 | 3 | 0 | MXXXXXXX | NNNNNNNNNN | | 1 | |
| 11 | B | | SHIP | 3 | 1 | 8 | 3 | 0 | MXXXXXXX | NNNNNNNNNN | MMI | 1 | |
| 12 | E | Y | SHIP | 3 | 1 | 8 | 3 | 0 | MXXXXXXX | NNNNNNNNNN | MMI | 1 | |
| 13 | B | | TEST | 2 | 1 | 9 | 4 | 0 | MXXXXXXX | NNNNNNNNNN | | | |
| 14 | E | Y | TEST | 2 | 1 | 9 | 4 | 0 | MXXXXXXX | NNNNNNNNNN | | | |
| 15 | B | | REJ | 4 | 1 | 9 | 4 | 0 | MXXXXXXX | NNNNNNNNNN | | | 8 |
| 16 | E | Y | REJ | 4 | 1 | 9 | 4 | 0 | MXXXXXXX | NNNNNNNNNN | | | 8 |
| 17 | B | | TEST | 2 | 1 | 17 | 5 | 0 | MXXXXXXX | NNNNNNNNNN | | | |
| 18 | E | | TEST | 2 | 1 | 17 | 5 | 0 | MXXXXXXX | NNNNNNNNNN | | | |
| 19 | B | | SHIP | 3 | 1 | 17 | 5 | 0 | MXXXXXXX | NNNNNNNNNN | MMI | 1 | |
| 20 | E | Y | SHIP | 3 | 1 | 17 | 5 | 0 | MXXXXXXX | NNNNNNNNNN | MMI | 1 | |
| 21 | B | | TEST | 2 | 1 | 17 | 6 | 0 | MXXXXXXX | NNNNNNNNNN | | 1 | |
| 22 | E | | TEST | 2 | 1 | 17 | 6 | 4 | XXXXXXXX | NNNNNNNNNN | | 1 | |
| 23 | B | | REJ | 4 | 1 | 17 | 6 | 4 | XXXXXXXX | NNNNNNNNNN | | 5 | 1 |
| 24 | E | Y | REJ | 4 | 1 | 17 | 6 | 4 | XXXXXXXX | NNNNNNNNNN | | 5 | 1 |

Figure 20-11 Request Execution Trace Report (Page 2 of 2)

The request execution trace report contains the following fields:

| | |
|---|---|
| REF-SEQ | A sequence number uniquely assigned to each execution trace record. |
| B/E | An indication of whether the report information is from the beginning or the end of the request. |
| OUT | An indication of whether any output to a report or subfile occurred during the request (applies at the end of a request only). |
| REQUEST | The executed request's name. |
| TYPE | The executed request's type. |
| REQ | The executed request's level number. Note that a non-subroutine request has a level number of one. A subroutine request referenced by a non-subroutine request has a level number of two. A subroutine request referenced by another subroutine request has a level number of three or higher. |
| MASTER | The processing sequence number of the master file record being processed by the executed request. |
| M4TRAN❹ | The processing sequence number of the transaction file record being processed by the executed request. |

Along with these fields, the settings of many VISION:Builder flags are listed. For more information regarding flag settings, refer to the *VISION:Builder Reference Summary*.

- ASTATUS the setting of the ASTATUS flag
- DELETE❹ the setting of the DELETE flag
- ECORD the setting of the ECORD flag
- MPASS the setting of the MISSPASS flag

- EOF the setting of the EOF flag
- LSTAT the setting of the LSTATUS flag
- RETURN the setting of the RETURNCD flag
- STRAN❹ the setting of the STRAN flag
- TRAN❹ the setting of the TRAN flag
- XTRAN❹ the setting of the XTRAN flag

If you look at the boxed items in the execution trace report, you will see that much information about the processing is presented. The first boxed item shows the flag settings at the end of the type 4 request REJ.

- The Y beneath the column heading OUT indicates that output is produced by that request for the fifth master file record and the first transaction file record.
- The M in the first byte of the ECORD flag indicates that a match occurred with the coordinated file.

The settings of the STRAN, TRAN, and XTRAN flags tell the transaction processing story.❹

- The M in the STRAN flag indicates that a match occurred at the first level of the master file.
- The 4 in the TRAN flag indicates that the transaction record was rejected.
- The 11 in the XTRAN flag states that a match action failed on a non-key field in the master file.

The second boxed item shows settings for the creation of a segment.

- The request is SHIP; it is a type 3 request processing the 17th master file record and the fifth transaction file record.
- A match has occurred against the coordinated file.

The STRAN and TRAN flag settings tell a different story this time.❹

- The STRAN flag indicates that matches have occurred against segments in the first two levels of the master file, while a segment has been inserted at the third level.
- The TRAN flag confirms that the master file record was updated.

## The GDBI Mapping Request Execution Trace Report

A GDBI mapping request execution trace report is also provided by the program analyzer's supplied requests. GDBI is described in the GDBI Programming Manual. This report shows the flow of mapping request events and the values of GDBI fields and flags at each request's beginning and ending point. Figure 20-12 depicts a GDBI mapping request execution trace report.

```
JAN 15, 1995                    MAPPING REQUEST EXECUTION TRACE FOR PROCESS , 01/15/95, 14.32.45                    PAGE  2
-------------------------------------------------------------------------------------------------------------------------------
REF-SEQ B/E OUT REQUEST  TYPE REQ COMMAND   DDNAME   FDNAME    FILE     FILEID  MODE SEG-LVL SEGNAME  MSTATUS
-------------------------------------------------------------------------------------------------------------------------------
   53   B      CUSTREQ   G    1  GETNEXT   M4CORD1  ORDRBILL M4CORD1           SR   01-1   CUSTOMER
   54   E      CUSTREQ   G    1  GETNEXT   M4CORD1  ORDRBILL M4CORD1           SR   01-1   CUSTOMER
   55   B      ORDREQ    G    1  GETFIRST  M4CORD1  ORDRBILL M4CORD1           SR   05-2   ORDER
   56   E      ORDREQ    G    1  GETFIRST  M4CORD1  ORDRBILL M4CORD1           SR   05-2   ORDER
   57   B      ORDREQ    G    1  GETNEXT   M4CORD1  ORDRBILL M4CORD1           SR   05-2   ORDER
   58   E      ORDREQ    G    1  GETNEXT   M4CORD1  ORDRBILL M4CORD1           SR   05-2   ORDER
   59   B      ORDREQ    G    1  GETNEXT   M4CORD1  ORDRBILL M4CORD1           SR   05-2   ORDER
   60   B      GDBISUB   GS   2  GETNEXT   M4CORD1  ORDRBILL M4CORD1           SR   05-2   ORDER    NFOUND
   61   E   Y  GDBISUB   GS   2  GETNEXT   M4CORD1  ORDRBILL M4CORD1           SR   05-2   ORDER    NFOUND
   62   E      ORDREQ    G    1  GETNEXT   M4CORD1  ORDRBILL M4CORD1           SR   05-2   ORDER    NFOUND
   63   B      CUSTREQ   G    1  GETNEXT   M4CORD1  ORDRBILL M4CORD1           SR   01-1   CUSTOMER
   64   E      CUSTREQ   G    1  GETNEXT   M4CORD1  ORDRBILL M4CORD1           SR   01-1   CUSTOMER NFOUND
   77   B      TERM      G    1  TERM      M4CORD1  ORDRBILL M4CORD1           SR   01-1
   78   E   Y  TERM      G    1  TERM      M4CORD1  ORDRBILL M4CORD1           SR   01-1
```

Figure 20-12 GDBI Mapping Request Execution Trace Report

The GDBI mapping request execution trace report contains the following information:

| | |
|---|---|
| REF-SEQ | A sequence number uniquely assigned to each execution trace record. |
| B/E | An indication of whether the report information is from the beginning or the end of the request. |
| OUT | An indication of whether any output to a report or subfile occurred during the request (applies at the end of a request only). |
| REQUEST | The executed request's name. |
| TYPE | The executed request's type. |
| REQ | The executed request's level number. Note that a non-subroutine request has a level number of one. A subroutine request referenced by a non-subroutine request has a level number of two. A subroutine request referenced by another subroutine request has a level number of three or higher. |
| COMMAND | The GDBI I/O command being executed by that request. |
| DDNAME | The VISION:Builder DTF/DD name for the mapped file (M4OLD, M4CORDn, and so on). |
| FDNAME | The setting of the FDNAME flag. |
| FILE | The setting of the FILE flag. |
| FILEID | The setting of the FILEID flag. |

| | |
|---|---|
| MODE | The setting of the MODE flag. |
| SEG-LVL | The segment and level numbers of the segment being processed. |
| SEGNAME | The name of the segment being processed. |
| MSTATUS | The setting of the MSTATUS flag. |

The boxed items within the mapping request execution trace report indicate the processing occurring during the GDBISUB request.

- This request is a GDBI subroutine request as indicated by the GS under the TYPE column heading. It is called by ORDREQ, because that request appeared immediately before GDBISUB in this report.
- The 2 in the REQ column indicates that this request has been branched to by another request.
- The GETNEXT command is performed by this request against a coordinated file named ORDRBILL. The GETNEXT command affects the ORDER segment that has a segment number of five and is at the second level.
- The MSTATUS flag setting indicates that the GETNEXT command did not find a segment.

These were the specifications at the beginning of the request, as well as the end of the request. Note that the REF-SEQ numbers are not all consecutive. The missing numbers can be found in the regular request execution trace report for this application. The mapping request references are a subset of all the references for the application.

## Using PAL to Help Maintain a Program

PAL is a helpful aid in the maintenance of an existing program. For example, a programmer developed a number of programs to help manage the expected print cost increases for the production department. One such program generated invoices to help update and maintain the customer file. Assume you are a maintenance programmer and a few months later are assigned to this program. The original programmer has left the company. You have been asked to modify the program to extract different totals and adjust the sales tax values. The efficiency of the program has also been questioned and it is your duty to increase the processing efficiency of the program.

Your first task is to understand the processing that exists already. One way to accomplish this is to slowly work through the processing and field usage on your own. Another way would be to generate the PAL reports for that application and use them as a road map of the application.

## What Fields are Being Used?

The field name cross reference report shows what fields are being used. This report is illustrated in Figure 20-13.

```
JAN 15, 1995                    FIELD NAME CROSS REFERENCE FOR INVOICE , 01/15/95, 14.01.41                    PAGE  1
-------------------------------------------------------------------------------------------------------------------------------
.....FIELD.............  .....REFERENCES.....
QLF  NAME     TYPE SEG-LVL  REQUEST  TYPE  LIST-SEQ.....(AND REFERENCE TYPE)                                    TOTAL CNV.
-------------------------------------------------------------------------------------------------------------------------------
    CITY      C     1-1     PROC     M     000037B
    CUSTADDR  C     1-1     PROC     M     000058R1  000074F1
    CUSTNAME  C     1-1     PROC     M     000056R1  000072F1
    CUSTZIP   C     1-1     PROC     M     000040B
    INSTCONT  C    60-2     PROC     M     000057R1  000073F1
    ITEMNAME  C    50-3     PROC     M     000062R1
    ITEMORD   SG   20-2     PROC     M     000043A   000044B
    ITMPRICE  Z    50-3     PROC     M     *00046A   000069R1                                                    1
    STATE     C     1-1     PROC     M     000033A   000039B
F   TODAY     C             PROC     M     000092R2  000097R2
L   SALESTAX  TB            PROC     M     000033B
L   SHIPCOST  TB            PROC     M     000049B
T   CHR       C             EOF      E     000103B
T   CHR       C             INVNO    1     000009E   000012C   000015R1
T   CHR       C             PROC     M     000054R1  000070F1  000095R2
T   CITYSTAT  C             PROC     M     000025E   000037C   000038A   000038C   000039A   000039C   000040A   000040C
                                             000059R1  000075F1
T   FILR      C             PROC     M     000028E   000099R2
T   FIL3      C             PROC     M     000026E   000093R2
T   HNDLG     Z             PROC     M     000023E   000049C   *00052B   000067R1  000083F1  000094R2                1
T   ID        C             PROC     M     000027E   000088R2
T   INVOICE   Z             EOF      E     000104B
T   INVOICE   Z             INVNO    1     000010E   *00013C   000016R1                                            1
T   INVOICE   Z             PROC     M     *00031A   *00031C   000055R1  000070F1  000096R2                      2
T   LASTINV   C             EOF      E     000102E   000103CP  *00104CP  000106R1                               1
T   RATE      Z             PROC     M     000019E   000033C   *00034A   *00034B   *00036C   *00048B             4
T   SALESTAX  Z             PROC     M     000022E   *00048C   *00051B   000066R1  000082F1                      2
T   TOTAL     Z             PROC     M     000024E   *00032C   *00050A   *00050C   *00051A   *00051C   *00052A   *00052C   7
                                             000068R1  000084F1  000098R2
T   TOTPRICE  Z             PROC     M     000020E   *00046C   *00048A   *00050B   000065R1  000081F1              3
T   TOTWGT    Z             PROC     M     000021E   *00047C   000049A   000063R1  000086F1                      1
X   CODE      C             PROC     M     000029A   000041A
X   CUSTNO    CK            PROC     M     000089R2
X   ITEMSHIP  C             PROC     M     000043B   000044A   000061R1
X   ITEMWGT   Z             PROC     M     *00047A                                                             1
X   ORDERNO   C             PROC     M     000060R1  000077F1  000090R2
X   QTYSHIP   Z             PROC     M     *00046B   *00047B   000064R1  000086F1                              2
X   SHIPNO    C             PROC     M     000091R2
1             RK    1-1     INVNO    1     000011RD
1   INVOICE   CK    1-1     INVNO    1     000012BP  000013BP  000014R1
```

Figure 20-13 What Fields are Being Used?

With this report you know what fields are used, how they are used, and where they are used. Note that a number of different types of fields are used in the application: master file fields, a flag field, tables, temporary fields, transaction file fields,❹ and coordinated file fields. From this report, note that the master file, qualified by a blank, has at least three levels and four segment types. The report also shows that the application uses two tables (reference type of TB) named SALESTAX and SHIPCOST. The numeric conversion totals should give you a clue to some processing efficiency problems.

## What Files are Used or Created?

The file name cross reference report shows what files are used or created (see Figure 20-14).

```
JAN 15, 1995                          FILE REFERENCE FOR INVOICE , 01/15/95, 14.01.41                          PAGE  1
-------------------------------------------------------------------------------------------------------------------------------
FILE       DTF/DD    FD/AD   ARR   DBD     GDBI  USER I/O  ...EN-FILE..........(REFERENCES).......
NAME        NAME      NAME   QLF   NAME    MODE   MODULE     NAME   REC  REQUEST  TYP  LIST-SEQ.....
-------------------------------------------------------------------------------------------------------------------------------
M4OLD      *SAME*   CUSTOMER
M4NEW      *SAME*   CUSTOMER
M4TRAN     *SAME*
M4CORD1    *SAME*   INVNUM
M4SUBF1    *SAME*                                          INVNUM        EOF       E   00105E1
M4SUBF2    *SAME*                                          INVINFO       PROC      M   00087E2
M4REJCT    *SAME*
```

Figure 20-14 Input and Output Files

The following files are processed by the application.

*Input to the application:*

- old master file
- transaction file❹
- coordinated file

*Output by the application:*

- new master file❹
- subfile INVNUM
- subfile INVINFO)
- rejected transaction file❹

The report also shows which request created each subfile.

With an understanding of the field and file usage for the application, you are ready to look at an execution trace of the processing (see Figure 20-15).

```
JAN 15, 1995                        REQUEST EXECUTION TRACE FOR INVOICE , 01/15/95, 14.01.41                        PAGE  1
-------------------------------------------------------------------------------------------------------------------------------
REF-SEQ B/E OUT REQUEST  TYPE REQ  MASTER   M4TRAN  ASTATUS DELETE   ECORD   MPASS     EOF     LSTAT RETURN   STRAN   TRAN XTRAN
                                                                   123456789       OT123456789              123456789
-------------------------------------------------------------------------------------------------------------------------------
   1   B      INVNO    1    1        1        1          0    XXXXXXXXX      NNNNNNNNNNN
   2   E   Y  INVNO    1    1        1        1          0    XXXXXXXXX      NNNNNNNNNNN
   3   B      PROC     M    1       10        1          0    XXXXXXXXX      NNNNNNNNNNN              MMU        1
   4   E      PROC     M    1       10        1          0    XXXXXXXXX      NNNNNNNNNNN              MMU        1
   5   B      PROC     M    1       10        2          0    XXXXXXXXX      NNNNNNNNNNN              MMMU       1
   6   E   Y  PROC     M    1       10        2          0    XXXXXXXXX      NNNNNNNNNNN              MMMU       1
   7   B      PROC     M    1       13        3          0    XXXXXXXXX      NNNNNNNNNNN              MMU        1
   8   E      PROC     M    1       13        3          0    XXXXXXXXX      NNNNNNNNNNN              MMU        1
   9   B      PROC     M    1       13        4          0    XXXXXXXXX      NNNNNNNNNNN              MMMU       1
  10   E   Y  PROC     M    1       13        4          0    XXXXXXXXX      NNNNNNNNNNN              MMMU       1
  11   B      PROC     M    1       13        5          0    XXXXXXXXX      NNNNNNNNNNN              MMMU       1
  12   E   Y  PROC     M    1       13        5          0    XXXXXXXXX      NNNNNNNNNNN              MMMU       1
  13   B      PROC     M    1       14        6          0    XXXXXXXXX      NNNNNNNNNNN              MMU        1
  14   E      PROC     M    1       14        6          0    XXXXXXXXX      NNNNNNNNNNN              MMU        1
  15   B      PROC     M    1       14        7          0    XXXXXXXXX      NNNNNNNNNNN              MMMU       1
  16   E   Y  PROC     M    1       14        7          0    XXXXXXXXX      NNNNNNNNNNN              MMMU       1
  17   B      PROC     M    1       15        8          0    XXXXXXXXX      NNNNNNNNNNN              MMU        1
  18   E      PROC     M    1       15        8          0    XXXXXXXXX      NNNNNNNNNNN              MMU        1
  19   B      PROC     M    1       15        9          0    XXXXXXXXX      NNNNNNNNNNN              MMMU       1
  20   E   Y  PROC     M    1       15        9          0    XXXXXXXXX      NNNNNNNNNNN              MMMU       1
  21   B      PROC     M    1       15       10          0    XXXXXXXXX      NNNNNNNNNNN              MMMU       1
  22   E   Y  PROC     M    1       15       10          0    XXXXXXXXX      NNNNNNNNNNN              MMMU       1
  23   B      EOF      E    1       18       10          0    XXXXXXXXX      NNNNNNNNNNN
  24   E   Y  EOF      E    1       18       10          0    XXXXXXXXX      NNNNNNNNNNN
```

Figure 20-15 What Type of Processing is Taking Place

From this report, note that the INVNO request initiates the processing and is not accessed again, while the EOF request brings the processing to a close. A key feature of this report is its illustration of transaction processing.❹ The STRAN settings, which specify the type of transaction processing taking place at each level of the master file, indicate that updates of the third and fourth levels are occurring.

Look at the sequence numbers of the records on the transaction file and the master file records being accessed and note that the third level of the file is first updated with one transaction record and a different transaction record updates the fourth level. This method of update occurs for all master file records shown other than the first and last.

With the information provided by the reports, you as a maintenance programmer should be able to better understand the intricacies of the processing occurring within the application.

# Source that Generates Reports

## File Definitions

Figure 20-16 illustrates the commands necessary to produce the program analyzer reports.

```
CONTROL
FILE MASTER INPUT  NAME IGCPALVB  KEYS NONE
FILE CORD1  NAME IGCPALRS  USERREAD
FILE REPORT
WORK 1  NAME IGCPALWK
INCLUDE IGCPAL
```

Figure 20-16 Commands for the PAL Report Run

The M4PAOUT file is specified as an old master file with the IGCPALVB file definition associated with it. A working storage definition (IGCPALWK) is also provided for M4WORK01. The file definitions for both the master file and working storage are provided below with a discussion of their use. A user-coordinated file (M4CORD1) that contains your selection of cross reference and trace reports is also specified in the run control group. The file definition (IGCPALRS) for the coordinated file is illustrated in Figure 20-5.

Figure 20-17 illustrates a portion of the source statements used to define the master file.

```
------------------------------------------------------------------------------------------------------------------------
   FILE    STMT    FILE    DELETE GLOSSARY RECORD RECORD RECORDS BUFFER MAPPED      MAPPING         MAPPED
   NAME    TYPE    IDENT    FILE   LEVEL   FORMAT  SIZE  PER BLK  SIZE   FILE    INIT      TERM     RECSIZ
------------------------------------------------------------------------------------------------------------------------
(IGCPALVB) (FD)             (Y)
***** UE00  TYPE 0  FILE IGCPALVB HAS BEEN DELETED.
(IGCPALVB) (FD)                     (B)      (V)    ( 168)         ( 4096)
COMMENT *****                                                        *****
COMMENT ***** IGCPALVB FD = M4PAOUT (VARIABLE BLOCKED).             *****
COMMENT *****                                                        *****
------------------------------------------------------------------------------------------------------------------------
   FILE    STMT   FIELD   DEL SGMT LEV FIELD  FIELD FLD SEG FLD DEC CNT  OCCRS OUTPT EDIT   COLUMN HEADING
   NAME    TYPE   NAME    ETE NMBR NUM LOCATN LNGTH TYP KEY RND PLC SGMT NTIME EDIT  LNGH       TEXT
------------------------------------------------------------------------------------------------------------------------
(IGCPALVB) (L0) (PARECKEY)    ( 1) (1) (   1) (  4) (F) (1)
(IGCPALVB) (L0) (PASRTKEY)             (   5) ( 48) (C)
(IGCPALVB) (L1) (PASRTKEY)                                                                  (PRE-SORT KEY    )
(IGCPALVB) (L0) (PARECID )             (   5) (  1) (C)
(IGCPALVB) (L1) (PARECID )                                                                  (REC-ID          )
COMMENT *****                                                        *****
COMMENT ***** REC-ID=1 - HEADER RECORD.....                         *****
COMMENT *****                                                        *****
COMMENT *****  (SORT= REC-ID.)                                      *****
COMMENT *****                                                        *****
(IGCPALVB) (L0) (PAHDR   )             (   1) (100) (C)
(IGCPALVB) (L0) (PAPROD  )             (   6) (  8) (C)
.
.
.
(IGCPALVB) (L0) (PASEGNL6)             ( 161) (  4) (C)                              (04)
(IGCPALVB) (L1) (PASEGNL6)                                                                  (SEG-LVL         )
***** UN00  TYPE 1  SEGMENT  1 LEVEL 1 CONTAINS OVERLAPPING FIELDS.
PAEXTRC  OVERLAPS PAHDR   ,PAFILEX ,PAXREF  ,PABRNCH ,PARECKEY,PASRTKEY,PASORT2 ,PARECID ,PASAVE5 ,PASAVE2 ,PACNTL23
                  PACNTL4 ,PATOLBL ,PAPROD  ,PAREFSEQ,PAQUAL  ,PAFUC   ,PAFIELD ,PASTYPE ,PADTFDD ,PABEGEND,PAOUTPUT
                  PATREQNM,PARUNNME,PAFILENM,PATOREQT,PATOLSEQ,PATTYPE ,PARUNDTE,PATREQLV,PAOWN   ,PAREQ5  ,PAFRREQ
                  PARUNTME,PAREQT5 ,PALSEQ5 ,PAFRREQT,PAFRLSEQ,PAFILL1 ,PAXREQ  ,PAREFT5 ,PAFRUSEQ,PAM4FILE,PAMFCNT
                  PATRCNT ,PAXTYP  ,PAREFR  ,PADOSASS,PALSEQ2 ,PAASTAT ,PAFRPSEQ,PATOUSEQ,PACOLUMN,PAREFTYP,PAFDNAM5
                  PAFLDTYP,PACONDCD,PAMAPFLD,PASEGNL2,PASEGN2 ,PADELETE,PAECORD ,PATOPSEQ,PADASH2 ,PASEGL2 ,PANCONV
                  PAARRQLF,PAMODE5 ,PANCONV1,PAUSEQ2 ,PADBDNAM,PAEOF   ,PAUSERIO,PALSTAT ,PAENREC ,PAUSEQ5 ,PAMSPASS
.
.
.
PAEOF    OVERLAPS PAUSERIO
 PAUSERIO OVERLAPS PALSTAT
 PALSTAT  OVERLAPS PAENREC ,PAUSEQ5
 PAUSEQ5  OVERLAPS PAMSPASS,PARETCDE,PAROW
```

Figure 20-17 IGCPALVB File Definition

The file definition for the master file is named IGCPALVB and defines a variable blocked record. This record in turn is overdefined a number of times to accommodate the six types of records output to M4PAOUT. The record ID is a single byte field that contains values from one to six. Each value corresponds to a different record type.

One set of fields is used for the header record (record ID of 1) that contains the product name and run name, date, and time.

A second set maps the field name and constant/pattern cross reference information (record IDs of 2 and 3). These fields contain the information output on either of those reports.

A third set of overdefinitions maps the information necessary for the branch location cross reference (record ID of 4) report.

A fourth set of overdefinitions applies to the file name cross reference information (record ID of 5).

A fifth set applies to the fields in the execution trace reports (record ID of 6). Within these overdefinitions, the particular fields that apply to the (GDBI) mapping request execution trace report are defined.

A portion of the source statements for the definition of the working storage area are shown in Figure 20-18.

Note that Figure 20-17 contains column heading text (L1 statements) that describes each field. For processing reasons, this information is *not* contained on the working storage definition provided by VISION:Builder, as shown in Figure 20-18.

```
-------------------------------------------------------------------------------------------------------------------
   FILE    STMT    FILE    DELETE GLOSSARY RECORD RECORD RECORDS BUFFER MAPPED      MAPPING         MAPPED
   NAME    TYPE    IDENT    FILE   LEVEL   FORMAT  SIZE  PER BLK  SIZE   FILE    INIT      TERM     RECSIZ
-------------------------------------------------------------------------------------------------------------------
(IGCPALWK) (FD)             (Y)
***** UF00  TYPE 0  FILE IGCPALWK HAS BEEN DELETED.
(IGCPALWK) (FD) (IGCPAL  )          (B)      (F)    ( 600)
COMMENT *****                                                          *****
COMMENT ***** IGCPALWK FD = WORKING STORAGE FOR REPORTING.             *****
COMMENT *****                                                          *****
-------------------------------------------------------------------------------------------------------------------
   FILE    STMT   FIELD   DEL SGMT LEV FIELD  FIELD FLD SEG FLD DEC CNT  OCCRS OUTPT EDIT   COLUMN HEADING
   NAME    TYPE   NAME    ETE NMBR NUM LOCATN LNGTH TYP KEY RND PLC SGMT NTIME EDIT  LNGH        TEXT
-------------------------------------------------------------------------------------------------------------------
(IGCPALWK) (L0) (PAHDR   )             (   1) (100) (C) (1)
(IGCPALWK) (L0) (PAPROD  )             (   6) (  8) (C)
(IGCPALWK) (L0) (PARUNNME)             (  14) (  8) (C)
.
.
.
(IGCPALWK) (L0) (PAREFT5 )             ( 538) (  2) (C)
(IGCPALWK) (L0) (PAM4FILE)             ( 540) (  8) (C)
(IGCPALWK) (L0) (PADOSASS)             ( 548) (  6) (C)
(IGCPALWK) (L0) (PAFDNAM5)             ( 554) (  8) (C)
(IGCPALWK) (L0) (PAARRQLF)             ( 562) (  1) (C)
(IGCPALWK) (L0) (PAMODE5 )             ( 563) (  2) (C)
(IGCPALWK) (L0) (PADBDNAM)             ( 565) (  8) (C)
(IGCPALWK) (L0) (PAUSERIO)             ( 573) (  8) (C)
(IGCPALWK) (L0) (PAENREC )             ( 581) (  1) (C)
(IGCPALWK) (L0) (PAUSEQ5 )             ( 582) (  8) (C)
(IGCPALWK) (L0) (WKREFR-5)             ( 590) ( 10) (C)


***** UN00  TYPE 1  SEGMENT  1 LEVEL 1 CONTAINS OVERLAPPING FIELDS.
PAHDR    OVERLAPS PAPROD  ,PARUNNME,PARUNDTE,PARUNTME,PAFILL1
SELECTS  OVERLAPS XREFNAME,XREFC/P ,XREFBRAN,XREFFILE,TRAC-REG,TRAC-MAP
PAXREF   OVERLAPS PACNTL2 ,PAQUAL2 ,PAFIELD2,PAXREQ2 ,PAXTYP2 ,PAREFR  ,PALSEQ2 ,PAREFTYP,PAFLDTYP,PAMAPFLD,PASEGNL2
                  PASEGN2 ,PADASH2 ,PASEGL2 ,PANCONV ,PANCONVI,PAUSEQ2
PACNTL2  OVERLAPS PAQUAL2 ,PAFIELD2,PAXREQ2 ,PAXTYP2
PAREFR   OVERLAPS PALSEQ2 ,PAREFTYP
PASEGNL2 OVERLAPS PASEGN2 ,PADASH2 ,PASEGL2
PACNTL3  OVERLAPS PAQUAL3 ,PAFIELD3,PAXREQ3 ,PAXTYP3
PACNTL4  OVERLAPS PATOLBL ,PATOREQT,PATOLSEQ,PAFRREQ ,PAFRREQT
PASAVE5  OVERLAPS PAFUC   ,PASTYPE ,PADTFDD ,PAFILENM,PAREQ5  ,PAREQT5 ,PALSEQ5 ,PAREFT5 ,PAM4FILE,PADOSASS,PAFDNAM5
                  PAARRQLF,PAMODE5 ,PADBDNAM,PAUSERIO,PAENREC ,PAUSEQ5
```

Figure 20-18 Working Storage Definition

The working storage definition contains a smaller number of fields than the master file. These fields are used primarily for processing the cross reference information and producing the reports you receive. You will see the use of the working storage fields in the next sections where the fixed format requests that generate the reports are described.

## The Requests

The two initial requests, IGCPALFT and IGCPALRS, shown in Figure 20-19, perform some basic processing to prepare for the processing of the M4PAOUT records themselves.

```
                                          ****************************
                                          *  REQUEST NAME - IGCPALFT  *
                                          *  INPUT STREAM REQUEST     *
                                          ****************************

-------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                                MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT PR
TYPE  DATE          REQUESTOR ID          ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP LP
-------------------------------------------------------------------------------------------------------------------------------
(ER)        (FIRST TIME ROUTINE.        )(   1)
-------------------------------------------------------------------------------------------------------------------------------
STMT SEQ  LOG CON .....OPERAND-A.....  OPER  .............OPERAND-B.............   ......RESULT......  PARTIAL FIELD
TYPE NO.  LEV CTR QLF FIELD   SEG-LVL  ATN   QLF FIELD   SEG-LVL  (OR CONSTANT)   QLF FIELD  SEG-LVL  STRT NUMB OPR
-------------------------------------------------------------------------------------------------------------------------------
(PR) (100)                             (R )   (C,YYYYYY                        ) (W,SELECTS )
(PR) (120)                             (R )   (C,@                             ) (W,ATAT    )



                                          ****************************
                                          *  REQUEST NAME - IGCPALRS  *
                                          *  INPUT STREAM REQUEST     *
                                          ****************************

-------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                                MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT PR
TYPE  DATE          REQUESTOR ID          ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP LP
-------------------------------------------------------------------------------------------------------------------------------
(ER)        (REPORTING SELECTION.       )(   1)
-------------------------------------------------------------------------------------------------------------------------------
STMT SEQ  LOG CON .....OPERAND-A.....  OPER  .............OPERAND-B.............   ......RESULT......  PARTIAL FIELD
TYPE NO.  LEV CTR QLF FIELD   SEG-LVL  ATN   QLF FIELD   SEG-LVL  (OR CONSTANT)   QLF FIELD  SEG-LVL  STRT NUMB OPR
-------------------------------------------------------------------------------------------------------------------------------
(PR) (100)                             (RD)   (1)
(PR) (120)         (1,REP-SEL ) 01-1   (EQ)   (C,IGCPAL                        )
(PR) (140)                             (NS)     (OUTPUT                       )
(PR) (160)                             (R )   (1,SELECTS ) 01-1                 (W,SELECTS )
-------------------------------------------------------------------------------------------------------------------------------
 ST SUM V  8  PG  PG   I SP  MAXIMUM TB COLUMN  DT  PG  STRT LIN SL RH .......SUBFILE......... C E A S R S         C
 TY RPT S LPI WID HGT  M FRM LPP PGS TO HEADING POS POS PAGE NUM UB PN ENT    NAME    BLK/ FMT H F G T S U         O
 MP     P             G            LT TYP POS         NUM      ML TD REC           BUF      C C S L T M         N
-------------------------------------------------------------------------------------------------------------------------------
(E1)                                  (F)
-------------------------------------------------------------------------------------------------------------------------------
         SP  Q              E  N  S  D  C  S  T  C  C  M  M  A  G  D Q PCT/RAT     P                PARTIAL
STMT SEQ  BF  L  FIELD  SEG-  N  P  E  E  T  U  0  U  N  A  I  V  M  E L  FIELD   RA- C     PICTURE     STRT NO
TYPE NO.  COL F  NAME     LVL  D  R  Q  S  L  B  T  M  T  X  N  G  R  C F  NAME    TIO T      EDIT       CHR CHR
-------------------------------------------------------------------------------------------------------------------------------
(R1) (100)     (W,XREFNAME)
(R1) (120)     (W,XREFC/P )
(R1) (140)     (W,XREFFRAN)
(R1) (160)     (W,XREFFILE)
(R1) (180)     (W,TRAC-REG)
(R1) (20 )     (W,TRAC-MAP)
-------------------------------------------------------------------------------------------------------------------------------
STMT  SEQ
TYPE  NO.                                                TEXT
-------------------------------------------------------------------------------------------------------------------------------
(T1)  (020)                 (IGCPAL - REPORTS SELECTED FROM M4PAOUT.                 )
```

Figure 20-19 Initial Requests

Request IGCPALFT loads working storage by replacing two character strings into working storage fields that are used during the PAL report processing.

Request IGCPALRS uses the RD operator to access the user coordinated file. It checks the first field in the coordinated file for the character string IGCPAL. If found, the 1,SELECTS field is placed into the working storage field, W,SELECTS.

This request produces a report that lists the entries in the coordinated file that select the wanted reports. Figure 20-20 illustrates this report when all the program analyzer reports are selected.

```
                                IGCPAL - REPORTS SELECTED FROM M4PAOUT.                                    PAGE 1
-------------------------------------------------------------------------------------------------------------------------------
  XREFNAME  XREFC/P  XREFBRAN  XREFFILE  TRAC-REG  TRAC-MAP
-------------------------------------------------------------------------------------------------------------------------------
     Y         Y        Y         Y         Y         Y
```

Figure 20-20 Report Selection Report

An important controlling request named IGCPALDR is illustrated in Figure 20-21.

```
-------------------------------------------------------------------------------------------------------------------------
STMT REPORT                                  MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT PR
TYPE  DATE           REQUESTOR ID           ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP LP
-------------------------------------------------------------------------------------------------------------------------
(ER)        (DRIVER ROUTINE.            )
-------------------------------------------------------------------------------------------------------------------------
STMT SEQ  LOG CON .....OPERAND-A.....  OPER  ............OPERAND-B.............   ......RESULT......  PARTIAL FIELD
TYPE NO.  LEV CTR QLF FIELD   SEG-LVL  ATN   QLF FIELD   SEG-LVL  (OR CONSTANT)   QLF FIELD  SEG-LVL  STRT NUMB OPR
-------------------------------------------------------------------------------------------------------------------------
(PR) (020)           (PARECID ) 01-1   (EQ)   (C,1                              )
(PR) (030)                             (NS)     (200                            )
(PR) (040)                             (GO)     (SUB IGCPAL1                    )
(PR) (050)                             (GO)     (NEXT MASTER                    )
(PR) (200)           (PARECID ) 01-1   (EQ)   (C,2                              )
(PR) (210)     (A)  (W,XREFNAME)       (EQ)   (C,Y                              )
(PR) (220)                             (NS)     (300                            )
(PR) (230)                             (GO)     (SUB IGCPAL2                    )
(PR) (240)                             (GO)     (NEXT MASTER                    )
(PR) (300)           (PARECID ) 01-1   (EQ)   (C,3                              )
(PR) (310)     (A)  (W,XREFC/P )       (EQ)   (C,Y                              )
(PR) (320)                             (NS)     (400                            )
(PR) (330)                             (GO)     (SUB IGCPAL3                    )
(PR) (340)                             (GO)     (NEXT MASTER                    )
(PR) (400)           (PARECID ) 01-1   (EQ)   (C,4                              )
(PR) (410)     (A)  (W,XREFBRAN)       (EQ)   (C,Y                              )
(PR) (420)                             (NS)     (500                            )
(PR) (430)                             (GO)     (SUB IGCPAL4                    )
(PR) (440)                             (GO)     (NEXT MASTER                    )
(PR) (500)           (PARECID ) 01-1   (EQ)   (C,5                              )
(PR) (510)     (A)  (W,XREFFILE)       (EQ)   (C,Y                              )
(PR) (520)                             (NS)     (600                            )
(PR) (530)                             (GO)     (SUB IGCPAL5                    )
(PR) (540)                             (GO)     (NEXT MASTER                    )
(PR) (600)           (PARECID ) 01-1   (EQ)   (C,6                              )
(PR) (610)     (A)   (PATTYPE ) 01-1   (NE)   (C,G                              )                      (01)  (01)  (A)
(PR) (620)     (A)  (W,TRAC-REG)       (EQ)   (C,Y                              )
(PR) (630)                             (NS)     (700                            )
(PR) (640)                             (GO)     (SUB IGCPAL6                    )
(PR) (650)                             (GO)     (NEXT MASTER                    )
(PR) (700)           (PARECID ) 01-1   (EQ)   (C,6                              )
(PR) (710)     (A)   (PATTYPE ) 01-1   (EQ)   (C,G                              )                      (01)  (01)  (A)
(PR) (720)     (A)  (W,TRAC-MAP)       (EQ)   (C,Y                              )
(PR) (730)                             (NS)     (800                            )
(PR) (740)                             (GO)     (SUB IGCPAL7                    )
(PR) (800)                             (GO)     (NEXT MASTER                    )
```

Figure 20-21 Controlling Request IGCPALDR

Request IGCPALDR drives the report processing of the M4PAOUT file. As mentioned earlier, each record of the PAL file has a record ID associated with it. The value of that ID corresponds to the type of program analyzer record. The following table illustrates the relationship between record ID and value.

| Record ID | Value |
|---|---|
| 1 | M4PAOUT header record. |
| 2 | Field name cross reference record. |
| 3 | Constant/pattern cross reference record. |
| 4 | Branch location cross reference record. |
| 5 | File name cross reference record. |
| 6 | Request trace flow record. |

Request IGCPALDR checks the record ID and, if reports have been requested for that type of record, branches to the subroutine that produces that type of report. It does this by first checking the master file field PARECID that contains the ID. Before branching to the requests that produce reports, the value of certain working storage fields are checked for Ys. These Ys correspond to the coordinated file selection record.

Recall how the first request IGCPALFT placed 1,SELECTS in W,SELECTS. The fields used in this IGCPALDR request override W,SELECTS and their field names refer to the type of report selected by an entry in that position of the coordinated file. Thus, sequence number 200 checks PARECID for a value of two (field name cross reference record) and sequence number 210 checks if a field name cross reference report was requested (as indicated by a Y in W,XREFNAME). If both of the relational expressions are true, a branch to subroutine IGCPAL2 takes place. IGCPAL2 processes the field name cross reference records to prepare them for reporting.

A similar three step process takes place for the other record IDs. Check for a particular ID value. If that value is found, check if that report is selected. If both conditions are true, branch to the subroutine request that processes those types of records. This process is true for the request trace reports (ID = 6); however, a third condition is added to determine if the trace is for regular requests or mapping requests. Note that the trace report logic in sequence numbers 600-800.

When the record ID is one, the record is a header record and request IGCPAL1 places the header information from the master file in the corresponding working storage field, shown in Figure 20-22.

```
-------------------------------------------------------------------------------------------------------------------------
STMT REPORT                                  MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT PR
TYPE  DATE           REQUESTOR ID           ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP LP
-------------------------------------------------------------------------------------------------------------------------
(ER)        (HEADER RECORD.             )                                          (S)
-------------------------------------------------------------------------------------------------------------------------
STMT SEQ  LOG CON .....OPERAND-A.....  OPER  ............OPERAND-B.............   ......RESULT......  PARTIAL FIELD
TYPE NO.  LEV CTR QLF FIELD   SEG-LVL  ATN   QLF FIELD   SEG-LVL  (OR CONSTANT)   QLF FIELD  SEG-LVL  STRT NUMB OPR
-------------------------------------------------------------------------------------------------------------------------
(PR) (020)           (PARECID ) 01-1   (EQ)    (C,1                            )
(PR) (040)                             (NS)      (END                          )
(PR) (060)                             (R )      (PAHDR   ) 01-1                 (W,PAHDR   )
(PR) (080)                             (GO)      (NEXT MASTER                  )
```

Figure 20-22 Processing Header Report

## Creating the Field Cross Reference Report

When the record ID is two and a field cross reference report is selected, request IGCPAL2 processes the field cross reference report records and prepares them for reporting. Figure 20-23 illustrates the IGCPAL2 request.

```
-------------------------------------------------------------------------------------------------------------------------
STMT REPORT                                  MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT PR
TYPE  DATE           REQUESTOR ID           ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP LP
-------------------------------------------------------------------------------------------------------------------------
(ER)        (NAME XREF PROCESSING.      )                                          (S)
-------------------------------------------------------------------------------------------------------------------------
STMT   FIELD    FIELD FLD DEC OUTPT EDIT      INITIAL          LINE 1 OF        LINE 2 OF
TYPE   NAME     LNGTH TYP PLC EDIT  LGTH       VALUE         COLUMN HEADING   COLUMN HEADING
-------------------------------------------------------------------------------------------------------------------------
(TF) (VREFR-2 ) ( 9H) (V)           (79)
-------------------------------------------------------------------------------------------------------------------------
STMT SEQ  LOG CON .....OPERAND-A.....  OPER  ............OPERAND-B.............   ......RESULT......  PARTIAL FIELD
TYPE NO.  LEV CTR QLF FIELD   SEG-LVL  ATN   QLF FIELD   SEG-LVL  (OR CONSTANT)   QLF FIELD  SEG-LVL  STRT NUMB OPR
-------------------------------------------------------------------------------------------------------------------------
(PR) (020)          (W,PAXREF  )       (EQ)    (C)
(PR) (040)                             (GS)      (300                          )
(PR) (060)          (W,PACNTL2 )       (EQ)      (PACNTL23) 01-1
(PR) (080)                             (GS)      (400                          )
(PR) (100)                             (GO)      (SUB IGCPAL2P                 )
(PR) (300)                             (R )    (C)                               (T,VREFR-2 )
(PR) (320)                             (R )    (D,0                            ) (W,TOTCNV2 )
(PR) (340)                             (R )      (PAXREF  ) 01-1                 (W,PAXREF  )
(PR) (400)            (PAREFTYP) 01-1  (SS)    (W,BLANK   )                      (W,ATAT    )
(PR) (420)          (W,TOTCNV2 )       (+ )      (PANCONV ) 01-1                 (W,TOTCNV2 )
(PR) (440)          (T,VREFR-2 )       (C2)      (PANCONVI) 01-1                 (T,VREFR-2 )
(PR) (460)          (T,VREFR-2 )       (C0)      (PAREFR  ) 01-1                 (T,VREFR-2 )
(PR) (580)                             (GO)      (RETURN                       )
```

Figure 20-23 Processing the Field Name Cross Reference Records

The processing constructs a variable length field that contains all the sequence number references, which are combined with an at sign (@). Each sequence number reference is preceded by an asterisk (contained in PANCONVI field if a numeric conversion is required at that reference). This request also loads a field

with the number of numeric conversions performed. These fields, named T,VREFR-2 and W,TOTCNV2, respectively, are reported by the IGCPAL2P request shown in Figure 20-24.

```
-----------------------------------------------------------------------------------------------------------------------
STMT REPORT                                  MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT PR
TYPE  DATE          REQUESTOR ID            ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP LP
-----------------------------------------------------------------------------------------------------------------------
(ER)        (NAME XREF REPORTING.       )                                          (S)
-----------------------------------------------------------------------------------------------------------------------
STMT SEQ  LOG CON .....OPERAND-A.....  OPER  ............OPERAND-B.............   ......RESULT......  PARTIAL FIELD
TYPE NO.  LEV CTR QLF FIELD   SEG-LVL  ATN   QLF FIELD   SEG-LVL  (OR CONSTANT)   QLF FIELD  SEG-LVL  STRT NUMB OPR
-----------------------------------------------------------------------------------------------------------------------
(PR) (020)         (W,PAXREF  )        (EQ)   (C)
(PR) (040)                             (GS)     (RETURN                        )
(PR) (060)         (T,VREFR-2 )        (SS)   (W,ATAT    )                      (W,BLANK   )
-----------------------------------------------------------------------------------------------------------------------
          SP  Q                 E  N  S  D  C  S  T  C  C  M  M  A  G  D Q PCT/RAT      P                  PARTIAL
STMT SEQ  BF  L  FIELD   SEG-   N  P  E  E  T  U  0  U  N  A  I  V  M  E L  FIELD   RA- C      PICTURE     STRT NO
TYPE NO.  COL F  NAME      LVL  D  R  Q  S  L  B  T  M  T  X  N  G  R  C F  NAME    TIO T       EDIT       CHR CHR
-----------------------------------------------------------------------------------------------------------------------
(R1) (020)     (W,PAPROD  )         (Y)
(R1) (030)     (W,PARUNNME)         (Y)
(R1) (040)     (W,PARUNDTE)         (Y)                                                   (##/##/##,      )
(R1) (050)     (W,PARUNTME)         (Y)
(R1) (060) (01) (W,PAQUAL2 )
(R1) (070) (02) (W,PAFIELD2)                                                                                (01) (08)
(R1) (080) (02) (W,PAFLDTYP)
(R1) (090) (03) (W,PASEGNL2)
(R1) (100) (04) (W,PAXREQ2 )
(R1) (120) (02) (W,PAXTYP2 )
(R1) (140) (03) (T,VREFR-2 )
(R1) (160) (03) (W,TOTCNV2 )
-----------------------------------------------------------------------------------------------------------------------
STMT  SEQ
TYPE  NO.                                                  TEXT
-----------------------------------------------------------------------------------------------------------------------
(F1)  (020)                  (PAGE-TITLE SECTION ##                                    )
(F1)  (030)                  (2S ##                                                    )
(F1)  (040)                  (F.DATE# 19B W.PAPROD# 1B                                 )
(F1)  (050)                  (#FIELD NAME CROSS REFERENCE FOR #                        )
(F1)  (060)                  (W.PARUNNME# 1#, # W.PARUNDTE# W.PARUNTME#                )
(F1)  (070)                  (19B #PAGE# 2B F.PAGE# ##                                 )
(F1)  (080)                  (1S ##                                                    )
(F1)  (090)                  (COLUMN-HEADING SECTION ##                                )
(F1)  (100)                  (1S ##                                                    )
(F1)  (110)                  (132#-# ##                                                )
(F1)  (120)                  (#.....FIELD............. #                              )
(F1)  (130)                  (#.....REFERENCES........ # ##                            )
(F1)  (140)                  (#QLF  NAME    TYPE SEG-LVL   #                           )
(F1)  (150)                  (#REQUEST  TYPE  LIST-SEQ.....(AND REFERENCE TYPE) #      )
(F1)  (160)                  (47B #TOTAL CNV. # ##                                     )

(F1)  (180)                  (1S ##                                                    )
```

Figure 20-24 Reporting the Field Name Cross Reference Records

Request IGCPAL2P creates the field name cross reference report. The PR logic checks if any data is available for reporting. If PAXREF field is blank, no field cross references have been placed in working storage and you return to the request that branched to this request originally. If cross reference records exist, the at (@) signs in the variable length field are replaced with blanks by the scan and substitute (SS) operation.

Once this check takes place, the fields are ready for reporting. Formatted sectional reporting is used to construct the title and column heading sections of the report. With these sections created, the request reports the fields you see on the field name cross reference report in Figure 20-7.

## Creating the Constant/Pattern Cross Reference Report

When the record ID is three and the constant/pattern cross reference report is selected, request IGCPAL3, shown in Figure 20-25, processes the constant and pattern cross reference records and prepares them for reporting.

```
-----------------------------------------------------------------------------------------------------------------
STMT REPORT                                  MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT PR
TYPE  DATE           REQUESTOR ID           ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP LP
-----------------------------------------------------------------------------------------------------------------
(ER)        (C/P XREF PROCESSING.       )                                                  (S)
-----------------------------------------------------------------------------------------------------------------
STMT   FIELD    FIELD FLD DEC OUTPT EDIT      INITIAL          LINE 1 OF        LINE 2 OF
TYPE   NAME     LNGTH TYP PLC EDIT  LGTH       VALUE         COLUMN HEADING   COLUMN HEADING
-----------------------------------------------------------------------------------------------------------------
(TF) (VREFR-3 ) ( 9H) (V)           (69)
-----------------------------------------------------------------------------------------------------------------
STMT SEQ  LOG CON .....OPERAND-A.....  OPER  ............OPERAND-B.............   ......RESULT......  PARTIAL FIELD
TYPE NO.  LEV CTR QLF FIELD   SEG-LVL  ATN   QLF FIELD   SEG-LVL  (OR CONSTANT)   QLF FIELD  SEG-LVL  STRT NUMB OPR
-----------------------------------------------------------------------------------------------------------------
(PR) (020)          (W,PACNTL3 )        (EQ)   (C)
(PR) (040)                              (GS)     (300                          )
(PR) (060)          (W,PACNTL3 )        (EQ)     (PACNTL23) 01-1
(PR) (080)                              (GS)     (400                          )
(PR) (100)                              (GO)     (SUB IGCPAL3P                 )
(PR) (300)                              (R )   (C)                                (T,VREFR-3 )
(PR) (320)                              (R )   (D,0                            ) (W,TOTCNV3 )
(PR) (340)                              (R )     (PACNTL23) 01-1                  (W,PACNTL3 )
(PR) (400)          (W,TOTCNV3 )        (+ )     (PANCONV ) 01-1                  (W,TOTCNV3 )
(PR) (410)          (T,VREFR-3 )        (C2)     (PANCONVI) 01-1                  (T,VREFR-3 )
(PR) (420)          (T,VREFR-3 )        (C0)     (PALSEQ2 ) 01-1                  (T,VREFR-3 )
(PR) (430)                              (GO)     (RETURN                       )
```

Figure 20-25 Processing the Constant/Pattern Cross Reference Records

It builds a variable length temporary field that contains the list sequence numbers for each reference, which are combined with an at (@) sign. Each sequence number reference is preceded by an asterisk (contained in PANCONVI if a numeric conversion is required at that reference). This request checks a control field named PACNTL23 containing the qualifier, the field name, and the request name and type. If these fields do not change, it is the same type of reference and the references are combined into the variable length temporary field. If the control field changes, it branches to the reporting request IGCPAL3P shown in Figure 20-26.

```
-----------------------------------------------------------------------------------------------------------------
STMT REPORT                                  MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT PR
TYPE  DATE           REQUESTOR ID           ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP LP
-----------------------------------------------------------------------------------------------------------------
(ER)        (C/P XREF REPORTING         )                                                  (S)
-----------------------------------------------------------------------------------------------------------------
STMT SEQ  LOG CON .....OPERAND-A..... OPER   ............OPERAND-B.............   ......RESULT......  PARTIAL FIELD
TYPE NO.  LEV CTR QLF FIELD   SEG-LVL  ATN   QLF FIELD   SEG-LVL  (OR CONSTANT)   QLF FIELD  SEG-LVL  STRT NUMB OPR
-----------------------------------------------------------------------------------------------------------------
(PR) (020)          (T,VREFR-3 )        (EQ)   (C)
(PR) (040)                              (GS)     (RETURN                       )
-----------------------------------------------------------------------------------------------------------------
          SP  Q                  E  N  S  D  C  S  T  C  C  M  M  A  G  D Q PCT/RAT      P                  PARTIAL
STMT SEQ  BF  L  FIELD   SEG-   N  P  E  E  T  U  0  U  N  A  I  V  M  E L  FIELD   RA- C      PICTURE     STRT NO
TYPE NO.  COL F  NAME      LVL  D  R  Q  S  L  B  T  M  T  X  N  G  R  C F  NAME    TIO T       EDIT       CHR CHR
-----------------------------------------------------------------------------------------------------------------
(R1) (020)     (W,PAPROD  )         (Y)
(R1) (030)     (W,PARUNNME)         (Y)
(R1) (040)     (W,PARUNDTE)         (Y)                                                    (##/##/##,      )
(R1) (050)     (W,PARUNTME)         (Y)
(R1) (060) (01) (W,PAQUAL3 )
(R1) (070) (02) (W,PAFIELD3)
(R1) (080) (02) (W,PAXREQ3 )
(R1) (090) (02) (W,PAXTYP3 )
(R1) (100) (03) (T,VREFR-3 )
(R1) (110) (04) (W,TOTCNV3 )
-----------------------------------------------------------------------------------------------------------------
```

Figure 20-26 Reporting the Constant/Pattern Cross Reference Records (Page 1 of 2)

```
STMT  SEQ
TYPE  NO.                                                  TEXT
-----------------------------------------------------------------------------------------------------------
(F1)  (020)                  (PAGE-TITLE SECTION ##                                          )
(F1)  (030)                  (2S ##                                                          )
(F1)  (040)                  (F.DATE# 16B W.PAPROD# 1B                                       )
(F1)  (050)                  (#CONSTANT/PATTERN CROSS REFERENCE FOR #                        )
(F1)  (060)                  (W.PARUNNME# 1,  W.PARUNDTE# W.PARUNTME# ##                     )
(F1)  (070)                  (15B #PAGE# 2B F.PAGE# ##                                       )
(F1)  (080)                  (1S ##                                                          )
(F1)  (090)                  (COLUMN-HEADING SECTION ##                                      )
(F1)  (100)                  (1S ##                                                          )
(F1)  (110)                  (132#-# ##                                                      )
(F1)  (120)                  (#.....CONSTANT OR PATTERN........... #                         )
(F1)  (130)                  (#..... REFERENCES........ # ##                                 )
(F1)  (140)                  (#QLF  VALUE                          #                         )
(F1)  (150)                  (#REQUEST  TYPE  LIST-SEQ.....(AND REFERENCE TYPE) #            )
(F1)  (160)                  (37B #TOTAL CNV.# ##                                            )
(F1)  (170)                  (132#-# ##                                                      )
(F1)  (180)                  (1S ##                                                          )
```

Figure 20-26 Reporting the Constant/Pattern Cross Reference Records (Page 2 of 2)

Request IGCPAL3P evaluates the temporary field named VREFR-3 to see if any constant/pattern references were made. If so, the report is produced. This report uses the formatted sectional reporting capabilities to format the Page-Title and Column-Heading sections and report the appropriate fields.

## Creating the Branch Location Cross Reference Report

When the record ID is four and a branch location cross reference report is selected, request IGCPAL4 processes the branch location cross reference report records and prepares them for reporting. Figure 20-27 illustrates the IGCPAL4 request.

```
-----------------------------------------------------------------------------------------------------------
STMT REPORT                                  MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT PR
TYPE  DATE          REQUESTOR ID            ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP LP
-----------------------------------------------------------------------------------------------------------
(ER)        (BRANCH XREF PROCESSING.    )                                          (S)
-----------------------------------------------------------------------------------------------------------
STMT   FIELD    FIELD FLD DEC OUTPT EDIT      INITIAL          LINE 1 OF        LINE 2 OF
TYPE   NAME     LNGTH TYP PLC EDIT  LGTH       VALUE         COLUMN HEADING   COLUMN HEADING
-----------------------------------------------------------------------------------------------------------
(TF)  (VREFR-4 )  ( 9H)  (V)           (87)
-----------------------------------------------------------------------------------------------------------
STMT SEQ  LOG CON .....OPERAND-A.....  OPER  ............OPERAND-B.............   ......RESULT......  PARTIAL FIELD
TYPE NO. LEV CTR  QLF FIELD   SEG-LVL  ATN   QLF FIELD   SEG-LVL  (OR CONSTANT)   QLF FIELD  SEG-LVL  STRT NUMB OPR
-----------------------------------------------------------------------------------------------------------
(PR) (020)          (W,PACNTL4 )        (EQ)    (C)
(PR) (040)                              (GS)      (300                          )
(PR) (060)          (W,PACNTL4 )        (EQ)      (PACNTL4 ) 01-1
(PR) (080)                              (GS)      (400                          )
(PR) (100)                              (GO)      (SUB IGCPAL4P                 )
(PR) (300)                              (R )    (C)                                (T,VREFR-4 )
(PR) (320)                              (R )      (PACNTL4 ) 01-1                  (W,PACNTL4 )
(PR) (400)          (T,VREFR-4 )        (C2)      (PAFRLSEQ) 01-1                  (T,VREFR-4 )
(PR) (420)                              (GO)      (RETURN                       )
```

Figure 20-27 Processing the Branch Location Cross Reference Records

The request constructs a variable length field containing the sequence numbers of all branch location references and branches to IGCPAL4P to report the appropriate fields for the report. Request IGCPAL4P is shown in Figure 20-28.

```
-------------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                                MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT PR
TYPE  DATE          REQUESTOR ID          ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP LP
-------------------------------------------------------------------------------------------------------------------------------------
(ER)         (BRANCH XREF REPORTING.     )                                        (S)
-------------------------------------------------------------------------------------------------------------------------------------
STMT SEQ  LOG CON .....OPERAND-A..... OPER  .............OPERAND-B.............  ......RESULT......  PARTIAL FIELD
TYPE NO. LEV CTR  QLF FIELD   SEG-LVL  ATN  QLF FIELD   SEG-LVL  (OR CONSTANT)  QLF FIELD  SEG-LVL  STRT NUMB OPR
-------------------------------------------------------------------------------------------------------------------------------------
(PR) (020)          (T,VREFR-4 )        (EQ)   (C)
(PR) (030)                              (NS)     (OUTPUT                      )
(PR) (040)                              (GO)     (RETURN                      )
-------------------------------------------------------------------------------------------------------------------------------------
         SP  Q               E  N  S  D  C  S  T  C  C  M  M  A  G  D Q PCT/RAT     P                PARTIAL
STMT SEQ BF  L  FIELD   SEG-  N  P  E  E  T  U  0  U  N  A  I  V  M  E L  FIELD   RA- C      PICTURE     STRT NO
TYPE NO. COL F  NAME      LVL  D  R  Q  S  L  B  T  M  T  X  N  G  R  C F  NAME    TIO T       EDIT       CHR CHR
-------------------------------------------------------------------------------------------------------------------------------------
(R1) (020)     (W,PAPROD  )         (Y)
(R1) (030)     (W,PARUNNME)         (Y)
(R1) (040)     (W,PARUNDTE)         (Y)                                                    (##/##/##,      )
(R1) (050)     (W,PARUNTME)         (Y)
(R1) (060) (00) (W,PATOLBL )
(R1) (070) (02) (W,PATOREQT)
(R1) (080) (03) (W,PATOLSEQ)
(R1) (090) (06) (W,PAFRREQ )
(R1) (100) (02) (W,PAFRREQT)
(R1) (110) (03) (T,VREFR-4 )
-------------------------------------------------------------------------------------------------------------------------------------
STMT  SEQ
TYPE  NO.                                                TEXT
-------------------------------------------------------------------------------------------------------------------------------------
(F1) (020)                  (PAGE-TITLE SECTION ##                                    )
(F1) (030)                  (2S ##                                                    )
(F1) (040)                  (F.DATE# 16B W.PAPROD# 1B                                 )
(F1) (050)                  (#BRANCH LOCATION CROSS REFERENCE FOR #                   )
(F1) (060)                  (W.PARUNNME# 1#, # W.PARUNDTE# W.PARUNTME#                )
(F1) (070)                  (17B #PAGE# 2B F.PAGE#  ##                                )
(F1) (080)                  (1S ##                                                    )
(F1) (090)                  (COLUMN-HEADING SECTION ##                                )
(F1) (100)                  (1S ##                                                    )
(F1) (110)                  (132#-# ##                                                )
(F1) (120)                  (#......BRANCH LOCATION..... #                            )
(F1) (130)                  (#.....FROM REFERENCE.....    # ##                        )
(F1) (140)                  (# REQUEST     TYPE  LIST-SEQ     #                       )
(F1) (150)                  (#REQUEST  TYPE  LIST-SEQ..... # ##                       )
(F1) (160)                  (132#-# ##                                                )
(F1) (170)                  (1S ##                                                    )
```

Figure 20-28 Reporting the Branch Location Cross Reference Records

Request IGCPAL4P evaluates the temporary field named VREFR-4 to see if any branch location references were made. If so, the report is produced. This report uses the formatted sectional reporting capabilities to format the Page-Title and Column-Heading sections and report the appropriate fields.

## Creating the File Name Cross Reference Report

When the record ID is five and a file name cross reference report is selected, request IGCPAL5 processes the file name cross reference report records and prepares them for reporting. Figure 20-29 illustrates the IGCPAL5 request.

```
-------------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                                MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT PR
TYPE  DATE          REQUESTOR ID          ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP LP
-------------------------------------------------------------------------------------------------------------------------------------
(ER)         (FILE XREF PROCESSING.      )                                        (S)
-------------------------------------------------------------------------------------------------------------------------------------
STMT   FIELD    FIELD FLD DEC OUTPT EDIT      INITIAL         LINE 1 OF       LINE 2 OF
TYPE   NAME     LNGTH TYP PLC EDIT  LGTH       VALUE        COLUMN HEADING  COLUMN HEADING
-------------------------------------------------------------------------------------------------------------------------------------
```

Figure 20-29 Processing the File Name Cross Reference Records (Page 1 of 2)

```
(TF) (VREFR-5 ) ( 5H) (V)           (45)
-------------------------------------------------------------------------------------------------------------------------
STMT SEQ  LOG CON .....OPERAND-A..... OPER  .............OPERAND-B.............    .....RESULT.....  PARTIAL FIELD
TYPE NO. LEV CTR  LF FIELD   SEG-LVL  ATN   QLF FIELD   SEG-LVL  (OR CONSTANT)    QLF FIELD  SEG-LVL  STRT NUMB OPR
-------------------------------------------------------------------------------------------------------------------------
(PR) (020)          (W,PAFUC    )       (EQ)   (D,0                          )
(PR) (040)                              (GS)     (300                        )
(PR) (060)          (W,PAFUC    )       (EQ)     (PAFUC    ) 01-1
(PR) (080)                              (NS)     (200                        )
(PR) (100)          (W,PADTFDD )        (NE)     (PADTFDD ) 01-1
(PR) (110)      (A) (W,PAFUC    )       (EQ)   (D,50                         )
(PR) (120)                              (GS)     (200                        )
(PR) (130)          (W,PAREQ5   )       (EQ)     (PAREQ5   ) 01-1
(PR) (140)      (O) (W,PAREQ5   )       (EQ)   (C)
(PR) (150)                              (GS)     (400                        )
(PR) (200)                              (GO)     (SUB IGCPAL5P               )
(PR) (300)                              (R )   (C)                             (T,VREFR-5 )
(PR) (320)                              (R )     (PASAVE5 ) 01-1               (W,PASAVE5 )
(PR) (400)          (T,VREFR-5 )        (C2)     (PALSEQ5 ) 01-1               (T,VREFR-5 )
(PR) (410)          (T,VREFR-5 )        (C0)     (PAREFT5 ) 01-1               (T,VREFR-5 )
(PR) (500)            (PASTYPE ) 01-1   (LT)   (D,1                          )
(PR) (510)                              (GS)     (RETURN                     )
(PR) (520)                              (R )     (PAFILENM) 01-1               (W,PAFILENM)
(PR) (530)                              (R )     (PAREQ5   ) 01-1              (W,PAREQ5   )
(PR) (540)                              (R )     (PAREQT5 ) 01-1               (W,PAREQT5 )
(PR) (550)                              (R )     (PAENREC ) 01-1               (W,PAENREC )
(PR) (560)            (PASTYPE ) 01-1   (EQ)   (D,2                          )
(PR) (570)                              (NS)     (RETURN                     )
(PR) (580)                              (R )     (PADTFDD ) 01-1               (W,PADTFDD )
(PR) (590)                              (R )     (PADOSASS) 01-1               (W,PADOSASS)
(PR) (600)                              (GO)     (RETURN                     )
```

Figure 20-29 Processing the File Name Cross Reference Records (Page 2 of 2)

The request constructs a variable length field containing the sequence numbers of references to files created using an En statement. It also loads the working storage fields corresponding to the fields of the file name cross reference record. After this processing and a check for multiple references to the same file, the request branches to the IGCPAL5P request shown in Figure 20-30.

```
-------------------------------------------------------------------------------------------------------------------------
STMT REPORT                                MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT PR
TYPE  DATE          REQUESTOR ID          ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP LP
-------------------------------------------------------------------------------------------------------------------------
(ER)        (FILE XREF REPORTING.      )                                          (S)
-------------------------------------------------------------------------------------------------------------------------
STMT SEQ  LOG CON .....OPERAND-A.....  OPER  .............OPERAND-B.............   .....RESULT.....  PARTIAL FIELD
TYPE NO.  LEV CTR QLF FIELD   SEG-LVL  ATN   QLF FIELD   SEG-LVL  (OR CONSTANT)   QLF FIELD  SEG-LVL  STRT NUMB OPR
-------------------------------------------------------------------------------------------------------------------------
(PR) (020)          (W,PASAVE5 )        (EQ)   (C)
(PR) (040)                              (GS)     (RETURN                     )
(PR) (060)          (W,PAM4FILE)        (EQ)   (W,PADTFDD )
(PR) (080)                              (NS)     (OUTPUT                     )
(PR) (100)                              (R )   (C, *SAME*                    ) (W,PADTFDD )
(PR) (120)          (W,PADOSASS)        (EQ)   (C)
(PR) (140)                              (NS)     (OUTPUT                     )
(PR) (160)                              (R )   (C, *NONE*                    ) (W,PADOSASS)
-------------------------------------------------------------------------------------------------------------------------
         SP Q              E N S D C S T C C M M A G D Q PCT/RAT     P               PARTIAL
STMT SEQ  BF L  FIELD  SEG- N P E E T U 0 U N A I V M E L FIELD   RA- C     PICTURE     STRT NO
TYPE NO.  COL F  NAME     LVL D R Q S L B T M T X N G R C F NAME   TIO T      EDIT       CHR CHR
-------------------------------------------------------------------------------------------------------------------------
(R1) (020)     (W,PAPROD  )         (Y)
(R1) (030)     (W,PARUNNME)         (Y)
(R1) (040)     (W,PARUNDTE)         (Y)                                          (##/##/##,      )
(R1) (050)     (W,PARUNTME)         (Y)
(R1) (060) (00) (W,PAM4FILE)
(R1) (070) (02) (W,PADTFDD )
(R1) (090) (02) (W,PAFDNAM5)
(R1) (100) (02) (W,PAARRQLF)
(R1) (110) (03) (W,PADBDNAM)
(R1) (130) (02) (W,PAMODE5 )
(R1) (140) (03) (W,PAUSERIO)
(R1) (150) (02) (W,PAFILENM)
(R1) (160) (02) (W,PAENREC )
(R1) (170) (03) (W,PAREQ5   )
(R1) (170) (02) (W,PAREQT5 )
(R1) (180) (03) (T,VREFR-5 )
-------------------------------------------------------------------------------------------------------------------------
STMT  SEQ
```

Figure 20-30 Reporting the File Name Cross Reference Records (Page 1 of 2)

```
TYPE  NO.                                          TEXT
-------------------------------------------------------------------------------------------------------
(F1) (020)                  (PAGE-TITLE SECTION ##                                        )
(F1) (030)                  (2S ##                                                        )
(F1) (040)                  (F.DATE# 25B W.PAPROD# 1B                                     )
(F1) (050)                  (#FILE REFERENCE FOR #                                        )
(F1) (060)                  (W.PARUNNME# 1#, # W.PARUNDTE# W.PARUNTME#                    )
(F1) (070)                  (25B #PAGE# 2B F.PAGE# ##                                     )
(F1) (090)                  (COLUMN-HEADING SECTION ##                                    )
(F1) (100)                  (1S ##                                                        )
(F1) (110)                  (132#-# ##                                                    )
(F1) (120)                  (#FILE       DTF/DD    FD/AD   ARR   DBD   #                  )
(F1) (130)                  (#  GDBI  USER I/O  ...EN-FILE#                               )
(F1) (140)                  (#..........(REFERENCES)...... # ##                           )
(F1) (150)                  (#NAME        NAME      NAME   QLF   NAME    #                 )
(F1) (160)                  (#  MODE   MODULE     NAME   REC#                             )
(F1) (160)                  (#  REQUEST  TYP  LIST-SEQ.....# ##                           )
(F1) (180)                  (132#-# ##                                                    )
(F1) (190)                  (1S ##                                                        )
```

Figure 20-30 Reporting the File Name Cross Reference Records (Page 2 of 2)

Request IGCPAL5P evaluates the working storage field named PASAVE5 to see if any file references were made. If so, the request checks to see if the VISION:Builder file name is the same as the DTF/DD name. If they are the same, VISION:Builder places the literal *SAME* into W,PADTFDD. If, under a VSE system, the ASSIGN is equal to blank, the literal *NONE* is placed into W,PADOSASS. This report uses the formatted sectional reporting capabilities to format the Page-Title and Column-Heading sections and report the appropriate fields.

## Creating the Execution Trace Reports

For the execution trace reports, no additional processing of the master file records is necessary to produce the reports. The subroutine requests output the master file fields that contain the appropriate data. Request IGCPAL6, shown in Figure 20-31, outputs the fields of the non-GDBI request execution trace reports, while IGCPAL7, shown in Figure 20-32, outputs the fields for the GDBI mapping request execution trace report.

```
-------------------------------------------------------------------------------------------------------
STMT REPORT                                  MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT PR
TYPE  DATE          REQUESTOR ID            ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP LP
-------------------------------------------------------------------------------------------------------
(ER)        (REG. TRACE REPORTING.      )                                          (S)
-------------------------------------------------------------------------------------------------------
          SP  Q                 E  N  S  D  C  S  T  C  C  M  M  A  G  D Q PCT/RAT      P                  PARTIAL
STMT SEQ  BF  L  FIELD   SEG-   N  P  E  E  T  U  0  U  N  A  I  V  M  E L  FIELD   RA- C      PICTURE    STRT NO
TYPE NO.  COL F  NAME     LVL   D  R  Q  S  L  B  T  M  T  X  N  G  R  C F  NAME    TIO T       EDIT      CHR CHR
-------------------------------------------------------------------------------------------------------
(R1) (020) (00)  (PAREFSEQ) 01-1
(R1) (030) (01)  (PABEGEND) 01-1
(R1) (040) (01)  (PAOUTPUT) 01-1
(R1) (050) (01)  (PATREQNM) 01-1
(R1) (060) (01)  (PATTYPE ) 01-1
(R1) (070) (01)  (PATREQLV) 01-1
(R1) (080) (01)  (PAMFCNT ) 01-1
(R1) (090) (01)  (PATRCNT ) 01-1
(R1) (100) (01)  (PAASTAT ) 01-1
(R1) (110) (01)  (PADELETE) 01-1
(R1) (120) (01)  (PAECORD ) 01-1
(R1) (130) (01)  (PAMSPASS) 01-1
(R1) (140) (01)  (PAEOF   ) 01-1
(R1) (150) (01)  (PALSTAT ) 01-1
(R1) (160) (01)  (PARETCDE) 01-1
(R1) (170) (01)  (PASTRAN ) 01-1
(R1) (180) (01)  (PATRAN  ) 01-1
(R1) (190) (01)  (PAXTRAN ) 01-1
COMMENT ***** 01 PAMSTAT                                                  *****
(R1) (210)     (W,PAPROD  )         (Y)
(R1) (220)     (W,PARUNNME)         (Y)
(R1) (230)     (W,PARUNDTE)         (Y)                                                  (##/##/##,      )
(R1) (240)     (W,PARUNTME)         (Y)
```

Figure 20-31 Reporting the Request Execution Trace Records (Page 1 of 2)

```
-------------------------------------------------------------------------------------------------------------
STMT  SEQ
TYPE  NO.                                                  TEXT
-------------------------------------------------------------------------------------------------------------
(F1)  (020)                  (PAGE-TITLE SECTION ##                                          )
(F1)  (030)                  (2S ##                                                          )
(F1)  (040)                  (F.DATE# 20B W.PAPROD# 1B #REQUEST EXECUTION TRACE FOR #        )
(F1)  (050)                  (W.PARUNNME# 1#, # W.PARUNDTE# W.PARUNTME#                      )
(F1)  (060)                  (21B #PAGE# 2B F.PAGE# ##                                       )
(F1)  (070)                  (1S ##                                                          )
```

Figure 20-31 Reporting the Request Execution Trace Records (Page 2 of 2)

```
-------------------------------------------------------------------------------------------------------------
STMT REPORT                                  MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT PR
TYPE  DATE          REQUESTOR ID            ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP LP
-------------------------------------------------------------------------------------------------------------
(ER)        (MAPPED TRACE REPORTING.    )                                          (S)                  (Y)
-------------------------------------------------------------------------------------------------------------
          SP  Q                 E  N  S  D  C  S  T  C  C  M  M  A  G  D Q PCT/RAT      P                  PARTIAL
STMT SEQ  BF  L  FIELD   SEG-   N  P  E  E  T  U  0  U  N  A  I  V  M  E L  FIELD   RA- C      PICTURE     STRT NO
TYPE NO.  COL F  NAME      LVL  D  R  Q  S  L  B  T  M  T  X  N  G  R  C F  NAME    TIO T       EDIT       CHR CHR
-------------------------------------------------------------------------------------------------------------
(R1) (020) (00)   (PAREFSEQ) 01-1
(R1) (030) (01)   (PABEGEND) 01-1
(R1) (040) (01)   (PAOUTPUT) 01-1
(R1) (050) (01)   (PATREQNM) 01-1
(R1) (060) (01)   (PATTYPE ) 01-1
(R1) (070) (01)   (PATREQLV) 01-1
(R1) (080) (01)   (PACOMMND) 01-1
(R1) (090) (01)   (PADDNAME) 01-1
(R1) (100) (01)   (PAFDNAM6) 01-1
(R1) (110) (01)   (PAFILE  ) 01-1
(R1) (120) (01)   (PAFILID6) 01-1
(R1) (130) (01)   (PAMODE  ) 01-1
(R1) (140) (01)   (PASEGNL6) 01-1
(R1) (150) (01)   (PASEGNAM) 01-1
(R1) (160) (01)   (PAMSTAT ) 01-1
(R1) (170)     (W,PAPROD  )          (Y)
(R1) (180)     (W,PARUNNME)          (Y)
(R1) (190)     (W,PARUNDTE)          (Y)                                                    (##/##/##,      )
(R1) (200)     (W,PARUNTME)          (Y)
-------------------------------------------------------------------------------------------------------------
STMT  SEQ
TYPE  NO.                                                  TEXT
-------------------------------------------------------------------------------------------------------------
(F1)  (020)                  (PAGE-TITLE SECTION ##                                          )
(F1)  (030)                  (2S ##                                                          )
(F1)  (040)                  (F.DATE# 16B W.PAPROD# 1B                                       )
(F1)  (040)                  (#MAPPING REQUEST EXECUTION TRACE FOR #                         )
(F1)  (050)                  (W.PARUNNME# 1#, # W.PARUNDTE# W.PARUNTME#                      )
(F1)  (060)                  (17B #PAGE# 2B F.PAGE# ##                                       )
(F1)  (070)                  (1S ##                                                          )
```

Figure 20-32 Reporting the GDBI Mapping Request Execution Trace Records

Request IGCPALEF, shown in Figure 20-33, is executed at end of file processing time after all records in the master file have been processed.

```
-------------------------------------------------------------------------------------------------------------
STMT REPORT                                  MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT PR
TYPE  DATE          REQUESTOR ID            ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP LP
-------------------------------------------------------------------------------------------------------------
(ER)        (EOF ROUTINE.               ) (   1)                                      (E)
-------------------------------------------------------------------------------------------------------------
STMT SEQ  LOG CON .....OPERAND-A.....  OPER  ............OPERAND-B.............   ......RESULT......  PARTIAL FIELD
TYPE NO.  LEV CTR QLF FIELD   SEG-LVL  ATN   QLF FIELD   SEG-LVL  (OR CONSTANT)   QLF FIELD  SEG-LVL  STRT NUMB OPR
-------------------------------------------------------------------------------------------------------------
(PR) (020)                             (GO)     (SUB IGCPAL2P                  )
(PR) (040)                             (GO)     (SUB IGCPAL3P                  )
(PR) (060)                             (GO)     (SUB IGCPAL4P                  )
(PR) (080)                             (GO)     (SUB IGCPAL5P                  )
```

Figure 20-33 End of File Processing

At end of file, this request branches to the subroutine requests that produce the actual cross reference reports to ensure the reporting of all values. Request IGCPAL2P produces the field cross reference report. Request IGCPAL3P generates the constant/pattern cross reference report. Request IGCPAL4P outputs the branch location cross reference report. Request IGCPAL5P produces the file cross reference report.

Chapter

# A Sample DB2 Applications

This appendix includes two sample DB2 application programs that print a glossary of columns in a DB2 table. In the first application, the user supplies a table name range and table creator ID using the PARM information in the JCL. The WH statement is used to select the specified tables from the SYSCOLUMNS table of the DB2 catalog. The sample application is preceded by the file definitions for the application.

```
   RUN     STMT    FILE    O N S D U T R A B S D S L  RB  R S M R S            START           END             SEQUENCE   LIST
   NAME    TYPE    NAME    L E E I P R E U U R L C S  PL  J O O F S            SEARCH          SEARCH          73 TO 80    SEQ
                           D W Q R D N P D F T M N T  TK  T P P O R
-------------------------------------------------------------------------------------------------------------------------------
(DB2SAMP1) (RC)                                                                                                                1

** CLSAA00  TYPE 0  DEFINITION SCOLUMNS IS BEING PROCESSED.                                                                *****
-------------------------------------------------------------------------------------------------------------------------------
   FILE    STMT    FILE    DELETE GLOSSARY RECORD RECORD RECORDS BUFFER MAPPED     MAPPING        MAPPED          SEQUENCE   LIST
   NAME    TYPE    IDENT    FILE   LEVEL   FORMAT  SIZE  PER BLK  SIZE   FILE    INIT      TERM     RECSIZ          73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(SCOLUMNS) (FD)             (R)      (N)      (S)                  (01024)                                                      2
COMMENT ***** ***************************************************   *****                                                      3
COMMENT ***** *                                                   *   *****                                                      4
COMMENT ***** *  SYSCOLUMNS TABLE                                 *   *****                                                      5
COMMENT ***** *                                                   *   *****                                                      6
COMMENT ***** ***************************************************   *****                                                      7
-------------------------------------------------------------------------------------------------------------------------------
   FILE    STMT  SEGMENT  DEL SGMT VRT            SUP SEG  KEY              CREATOR        EXTERNAL            SEQUENCE   LIST
   NAME    TYPE   NAME    ETE NMBR SEG            DPL KEY CHAR                ID          TABLE NAME          73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(SCOLUMNS) (LS) (SCOLUMNS)    ( 1)                                          (SYSDB2  ) (SYSCOLUMNS        )                8
-------------------------------------------------------------------------------------------------------------------------------
   FILE    STMT   FIELD   DEL SGMT LEV FIELD  FIELD FLD SEG FLD DEC CNT  OCCRS OUTPT EDIT   DATA BASE NAME/        SEQUENCE   LIST
   NAME    TYPE   NAME    ETE NMBR NUM LOCATN LNGTH TYP KEY RND PLC SGMT NTIME EDIT  LNGH   COLUMN HEADING         73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(SCOLUMNS) (L0) (COLTYPE )    ( 1) (1) (0001) (008) (C)                                    (COLTYPE              )              9
(SCOLUMNS) (L1) (COLTYPE )                                                                 (COLUMN               )             10
(SCOLUMNS) (L2) (COLTYPE )                                                                 (TYPE                 )             11
(SCOLUMNS) (L0) (LENGTH  )    ( 1) (1) (0009) (007) (C)                                    (LENGTH               )             12
(SCOLUMNS) (L1) (LENGTH  )                                                                 (COLUMN               )             13
(SCOLUMNS) (L2) (LENGTH  )                                                                 (LENGTH               )             14
(SCOLUMNS) (L0) (TNAME   )    ( 1) (1) (0016) (018) (C) (1)                                (TBNAME               )             15
(SCOLUMNS) (L1) (TNAME   )                                                                 (TABLE                )             16
(SCOLUMNS) (L2) (TNAME   )                                                                 (NAME                 )             17
(SCOLUMNS) (L0) (CREATOR )    ( 1) (1) (0034) (008) (C)                                    (TBCREATOR            )             18
(SCOLUMNS) (L1) (CREATOR )                                                                 (TABLE                )             19
(SCOLUMNS) (L2) (CREATOR )                                                                 (CREATOR              )             20
(SCOLUMNS) (L0) (NULLS   )    ( 1) (1) (0042) (001) (C)                                    (NULLS                )             21
(SCOLUMNS) (L1) (NULLS   )                                                                 (NULLS                )             22
(SCOLUMNS) (L2) (NULLS   )                                                                 (ALLOWED?             )             23
(SCOLUMNS) (L0) (CNAME   )    ( 1) (1)        (018) (V)                                    (NAME                 )             24
(SCOLUMNS) (L1) (CNAME   )                                                                 (COLUMN               )             25
(SCOLUMNS) (L2) (CNAME   )                                                                 (NAME                 )             26
(SCOLUMNS) (L0) (REMARKS )    ( 1) (1)        (254) (V)                              (20) (REMARKS              )             27
(SCOLUMNS) (L1) (REMARKS )                                                                 (COLUMN               )             28
(SCOLUMNS) (L2) (REMARKS )                                                                 (DESCRIPTION          )             29
** CLSDC02  TYPE 0  THE DEFINITION IS VALID AND HAS BEEN CATALOGED.                                                          *****

** CLSAA00  TYPE 0  DEFINITION OSPARMFD IS BEING PROCESSED.                                                                *****
-------------------------------------------------------------------------------------------------------------------------------
   FILE    STMT    FILE    DELETE GLOSSARY RECORD RECORD RECORDS BUFFER MAPPED     MAPPING        MAPPED          SEQUENCE   LIST
```

Figure A-1 Sample 1: File Definitions (Page 1 of 2)

```
   NAME    TYPE    IDENT    FILE   LEVEL   FORMAT  SIZE  PER BLK  SIZE   FILE    INIT      TERM     RECSIZ          73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(OSPARMFD) (FD)             (R)      (N)      (F)                 (   82)                                                    30
-------------------------------------------------------------------------------------------------------------------------------
   FILE    STMT  SEGMENT  DEL SGMT VRT            SUP SEG  KEY              CREATOR        EXTERNAL            SEQUENCE   LIST
   NAME    TYPE   NAME    ETE NMBR SEG            DPL KEY CHAR               ID          TABLE NAME           73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(OSPARMFD) (LS) (PARM01  )     ( 1)                                                                                        31
-------------------------------------------------------------------------------------------------------------------------------
   FILE    STMT   FIELD   DEL SGMT LEV FIELD  FIELD FLD SEG FLD DEC CNT  OCCRS OUTPT EDIT   DATA BASE NAME/          SEQUENCE   LIST
   NAME    TYPE   NAME    ETE NMBR NUM LOCATN LNGTH TYP KEY RND PLC SGMT NTIME EDIT  LNGH   COLUMN HEADING           73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(OSPARMFD) (LO) (PARMSIZE)     ( 1) (1) (   1) (  2) (F)                                                                   32
(OSPARMFD) (LO) (PARMDATA)     ( 1) (1) (   3) ( 80) (C) (1)                                                               33
(OSPARMFD) (LO) (FROM    )     ( 1) (1) (   3) ( 18) (C)                                                                   34
(OSPARMFD) (LO) (TO      )     ( 1) (1) (  21) ( 18) (C)                                                                   35
(OSPARMFD) (LO) (CREATOR )     ( 1) (1) (  39) (  8) (C)                                                                   36
** CLSUN00  TYPE 1  SEGMENT  1 LEVEL 1 CONTAINS OVERLAPPING FIELDS.                                                      *****
                    PARMDATA OVERLAPS FROM     TO       CREATOR                                                          *****
** CLSDC02  TYPE 0  THE DEFINITION IS VALID AND HAS BEEN CATALOGED.                                                       *****
```

Figure A-1 Sample 1: File Definitions (Page 2 of 2)

```
   RUN     STMT    FILE    O N S D U T R A B S D S L  RB  R S M R S          START          END          SEQUENCE   LIST
   NAME    TYPE    NAME    L E E I P R E U U R L C S  PL  J O O F S          SEARCH         SEARCH       73 TO 80    SEQ
                           D W Q R D N P D F T M N T  TK  T P P O R
-------------------------------------------------------------------------------------------------------------------------------
(DB2SAMP1) (RC)  (SCOLUMNS) (S) (U)      (S)      (#,T)           (R)                                                      1
-------------------------------------------------------------------------------------------------------------------------------
   RUN     STMT  PARAMETER                                                                                          SEQUENCE   LIST
   NAME    TYPE    NAME                              PARAMETER VALUE                                              73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(DB2SAMP1) (RP)  (DB2     )       (DB2T    INM4CALL                                  )                                    3
-------------------------------------------------------------------------------------------------------------------------------
   RUN     STMT  LOGICAL  LBL  DTF/DD     KEY      KEY      KEY     QUALIFIER  USER I/O  ARRAY      PASSWORD/  SEQUENCE   LIST
   NAME    TYPE  FILENAME      NAME     FIELD-1  FIELD-2  FIELD-3  ICF CRD DBI  MODULE   QUAL       AUTH.-ID  73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(DB2SAMP1) (RF) (OSPARMFD)    (M4LINK01)                                                                                  4
-------------------------------------------------------------------------------------------------------------------------------
   RUN     STMT   SEGMENT    FILE    WHERE                                                                          SEQUENCE   LIST
   NAME    TYPE    NAME      QUAL    PREDICATE                                                                      73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(DB2SAMP1) (WH)   (SCOLUMNS)  (O)    (TBCREATOR = :V.CREATOR AND                    )                                        5
(DB2SAMP1) (WH)   (SCOLUMNS)  (O)    (TBNAME BETWEEN :V.FROM AND :V.TO              )                                        6

                                          ******************************
                                          *  PROC NAME - MAIN          *
                                          *  INPUT STREAM PROCEDURE    *
                                          ******************************
-------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                              MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID        ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM       73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(ER)                                                                          (N)                                            7
    ***** ASL Procedure Follows - Using Translator Version x.xx
    ##PROC                                                                                                  (       )     8
    ;EACH MASTER-FILE-RECORD CONTROL PROCEDURE                                                               (       )     9
    ;                                                                                                       (       )    10
    ;  TEMPORARY FIELDS                                                                                     (       )    11
    ;                                                                                                       (       )    12
    BLANK:    FIELD C 1                                                                                     (       )    13
    TNAMEX:   FIELD C 6  INIT 'TABLE:'                                                                      (       )    14
    TABLES1:  FIELD C 8  INIT 'TABLE(S)'                                                                    (       )    15
    TABLES2:  FIELD C 10 INIT 'ALL TABLES'                                                                  (       )    16
    CREATED:  FIELD C 11 INIT 'CREATED BY'                                                                  (       )    17
    TOX:      FIELD C 3  INIT 'TO:'                                                                         (       )    18
    FROMX:    FIELD C 5  INIT 'FROM:'                                                                       (       )    19
    TABLENAM: FIELD C 18                                                                                    (       )    20
    ;                                                                                                       (       )    21
    CALL REPORT LISTNAME                                                                                    (       )    22
    ##PEND                                                                                                  (       )    23
    ***** End of ASL Procedure
```

Figure A-2 Sample 1: Application Code (Page 1 of 4)

```
                                        ******************************
                                        *  PROC NAME - EOF           *
                                        *  INPUT STREAM PROCEDURE    *
                                        ******************************

-------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                                MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID          ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM      73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(ER)                                                                          (E)                                          24
    ***** ASL Procedure Follows - Using Translator Version x.xx
    ##PROC                                                                                                    (       )   25
    ;END-OF-FILE CONTROL PROCEDURE                                                                            (       )   26
    ;                                                                                                         (       )   27
    CALL REPORT LISTPARM                                                                                      (       )   28
    ##PEND                                                                                                    (       )   29
    ***** End of ASL Procedure

                                        ******************************
                                        *  REQUEST NAME - LISTPARM   *
                                        *  INPUT STREAM REQUEST      *
                                        ******************************

-------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                                MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID          ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM      73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(ER) (TODAY )                                                                 (S)                                          30
COMMENT ***** PARM FIELD DATA REPORT                                  *****                                                31

-------------------------------------------------------------------------------------------------------------------------------
 ST SUM V  8  PG  PG   I SP  MAXIMUM TB COLUMN  DT  PG  STRT LIN SL RH .......SUBFILE........ C E A S R S N      C  SEQUENCE   LIST
 TY RPT S LPI WID HGT  M FRM LPP PGS TO HEADING POS POS PAGE NUM UB PN ENT    NAME   BLK/ FMT H F G T S U D      O  73 TO 80    SEQ
 MP     P            G            LT TYP POS         NUM      ML TD REC           BUF      C C S L T M S      N
-------------------------------------------------------------------------------------------------------------------------------
(E1)         ( 80)                   (F)                                                                                  32
-------------------------------------------------------------------------------------------------------------------------------
         SP  Q              E N S D C S T C C M M A G D Q PCT/RAT     P              PARTIAL   SEQUENCE   LIST
STMT SEQ BF  L  FIELD  SEG-  N P E E T U 0 U N A I V M E L  FIELD   RA- C     PICTURE    STRT NO  73 TO 80    SEQ
TYPE NO. COL F  NAME     LVL D R Q S L B T M T X N G R C F  NAME    TIO T      EDIT      CHR CHR
-------------------------------------------------------------------------------------------------------------------------------
(R1)         (V,FROM    )                                                                                                 33
(R1)         (V,TO      )                                                                                                 34
(R1)         (V,CREATOR )                                                                                                 35
-------------------------------------------------------------------------------------------------------------------------------
STMT  SEQ                                                                                                    SEQUENCE   LIST
TYPE  NO.                                  TEXT                                                             73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(T1)                      (                                  PARM FIELD INFOR)                                           36
(T1)                      (MATION            #                           )                                             37

                                        ******************************
                                        *  REQUEST NAME - LISTNAME   *
                                        *  INPUT STREAM REQUEST      *
                                        ******************************

-------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                                MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID          ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM      73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(ER) (TODAY )                                                                 (S)                                          38
COMMENT ***** DB2 TABLES GLOSSARY                                     *****                                                39

-------------------------------------------------------------------------------------------------------------------------------
 ST SUM V  8  PG  PG   I SP  MAXIMUM TB COLUMN  DT  PG  STRT LIN SL RH .......SUBFILE........ C E A S R S N      C  SEQUENCE   LIST
 TY RPT S LPI WID HGT  M FRM LPP PGS TO HEADING POS POS PAGE NUM UB PN ENT    NAME   BLK/ FMT H F G T S U D      O  73 TO 80    SEQ
 MP     P            G            LT TYP POS         NUM      ML TD REC           BUF      C C S L T M S      N
-------------------------------------------------------------------------------------------------------------------------------
(E1)         ( 80)                                                                                                        40
-------------------------------------------------------------------------------------------------------------------------------
         SP  Q              E N S D C S T C C M M A G D Q PCT/RAT     P              PARTIAL   SEQUENCE   LIST
STMT SEQ BF  L  FIELD  SEG-  N P E E T U 0 U N A I V M E L  FIELD   RA- C     PICTURE    STRT NO  73 TO 80    SEQ
TYPE NO. COL F  NAME     LVL D R Q S L B T M T X N G R C F  NAME    TIO T      EDIT      CHR CHR
-------------------------------------------------------------------------------------------------------------------------------
(R1)      (11) (T,TABLES2 )              (1) (P)                                                                           41
(R1)      ( 1) (T,FROMX   )              (1) (P)                                                                           42
(R1)      ( 1) (V,FROM    )              (1) (P)                                                                           43
(R1)      ( 1) (T,TOX     )              (1) (P)                                                                           44
(R1)      ( 1) (V,TO      )      (Y)     (1) (P)                                                                           45
(R1)           (T,BLANK   )      (Y)     (1) (P)                                                                           46
(R1)      (15) (T,TNAMEX  )              (1) (P)                                                                           47
(R1)           (N,TNAME   ) 01-1   (2)   (1) (P)                                                                           48
(R1)           (T,CREATED )              (1) (P)                                                                           49
(R1)           (V,CREATOR )         (1)  (1) (P)                                                                           50
(R1)           (N,CNAME   ) 01-1                                                                                           51
(R1)           (N,COLTYPE ) 01-1                                                                                           52
(R1)           (N,LENGTH  ) 01-1                                                                                           53
(R1)           (N,NULLS   ) 01-1                                                                                           54
(R1)           (N,REMARKS ) 01-1                                                                                           55
```

Figure A-2 Sample 1: Application Code (Page 2 of 4)

```
-------------------------------------------------------------------------------------------------------------------------------
STMT  SEQ                                                                                                  SEQUENCE   LIST
TYPE  NO.                                          TEXT                                                   73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(T1)                        (              VISION:BUILDER/DB2 GLOSSARY UTILITY - 2.0  )                                    56
(T1)                        (                  #                                      )                                    57
(T1)                                                                                                                        58
(T1)                        (                  #                                      )                                    59
(T1)                        (                      GLOBAL DB2 TABLES GLOSSARY         )                                    60
(T1)                        (                  #                                      )                                    61

                                      ********************************************
                                      *   SUMMARY OF GENERATED DB2 STATEMENTS   *
                                      ********************************************
-------------------------------------------------------------------------------------------------------------------------------
FILE>SEGMENT                  DB2 STATEMENT                                                  COMMENTS
-------------------------------------------------------------------------------------------------------------------------------
M4OLD>SCOLUMNS     STATEMENT  1
          SELECT COLTYPE,LENGTH,TBNAME,NULLS,NAME,REMARKS FROM              STANDARD SELECT FOR SEQUENTIAL READ OF ROOT SEGMENT.
          SYSDB2.SYSCOLUMNS X1 WHERE TBCREATOR = ? AND TBNAME BETWEEN       WHERE CLAUSE BUILT FROM: (WH) STATEMENT. DB2 CALL(S):
          ? AND ?                                                           1 PREPARE(S), 1 OPEN(S), AND 34 FETCH(ES).
          HOST VARIABLE(S) = V.CREATOR V.FROM V.TO

** MK4W204  TYPE 0  NUMBER OF MESSAGES PRINTED IS    0.
                    M4OLD    -       34 RECORDS INPUT
                    M4REPO   -       63 RECORDS OUTPUT
                    M4INPUT  -       61 RECORDS INPUT
                    M4LIST   -      226 RECORDS OUTPUT
                       3 TRACKS ASSIGNED TO M4LIB    --    2 TRACKS NOT FULL. LIBRARY DIRECTORY BLOCKING FACTOR IS  138.
** MK4W209  TYPE 0  1018624 BYTES OF MAIN STORAGE UNUSED DURING DECODING PHASE
                    1021952 BYTES OF MAIN STORAGE UNUSED DURING COMPILATION PHASE
                     634880 BYTES OF MAIN STORAGE UNUSED DURING PROCESSING PHASE
** MK4ED02  TYPE 0  REQUEST-LISTPARM  REPORT-1  REQUESTOR ID-

01/15/95                                                                PAGE 1
                              PARM FIELD INFORMATION
-------------------------------------------------------------------------------
        FROM                 TO          CREATOR
-------------------------------------------------------------------------------
  INM                 INN                ISPBJM1

** MK4FT03  TYPE 0  END OF REPORT.

** MK4ED02  TYPE 0  REQUEST-LISTNAME  REPORT-1  REQUESTOR ID-

01/15/95                                                                PAGE 1
               VISION:BUILDER/DB2 GLOSSARY UTILITY - 2.0
                          GLOBAL DB2 TABLES GLOSSARY
          ALL TABLES FROM: INM                 TO: INN
              TABLE:  INMADDR             CREATED BY   ISPBJM1
-------------------------------------------------------------------------------
       COLUMN         COLUMN   COLUMN    NULLS          COLUMN
        NAME           TYPE    LENGTH   ALLOWED?      DESCRIPTION
-------------------------------------------------------------------------------
  ADDRNO             CHAR           2     N
  CITYSTAT           CHAR          30     N
  EMPNO              CHAR           6     N
  STREET             CHAR          30     N

01/15/95                                                                PAGE 2
               VISION:BUILDER/DB2 GLOSSARY UTILITY - 2.0
                          GLOBAL DB2 TABLES GLOSSARY
          ALL TABLES FROM: INM                 TO: INN
              TABLE:  INMDATE             CREATED BY   ISPBJM1
-------------------------------------------------------------------------------
       COLUMN         COLUMN   COLUMN    NULLS          COLUMN
        NAME           TYPE    LENGTH   ALLOWED?      DESCRIPTION
-------------------------------------------------------------------------------
  BEGDATE            DATE           4     N
  EMPNO              CHAR           6     N
  ENDDATE            DATE           4     N
```

Figure A-2 Sample 1: Application Code (Page 3 of 4)

```
01/15/95                                                              PAGE 3
              VISION:BUILDER/DB2 GLOSSARY UTILITY - 2.0
                        GLOBAL DB2 TABLES GLOSSARY
          ALL TABLES FROM: INM                TO: INN
             TABLE:  INMDEPT              CREATED BY   ISPBJM1
-------------------------------------------------------------------------------
       COLUMN         COLUMN   COLUMN    NULLS          COLUMN
        NAME           TYPE    LENGTH   ALLOWED?      DESCRIPTION
-------------------------------------------------------------------------------
  ADMRDEPT            CHAR          3     Y
  DEPTNAME            CHAR         36     N
  DEPTNO              CHAR          3     Y
  MGRNO               CHAR          6     Y

01/15/95                                                              PAGE 4
              VISION:BUILDER/DB2 GLOSSARY UTILITY - 2.0
                        GLOBAL DB2 TABLES GLOSSARY
          ALL TABLES FROM: INM                TO: INN
             TABLE:  INMEMPL              CREATED BY   ISPBJM1
-------------------------------------------------------------------------------
       COLUMN         COLUMN   COLUMN    NULLS          COLUMN
        NAME           TYPE    LENGTH   ALLOWED?      DESCRIPTION
-------------------------------------------------------------------------------
  BRTHDATE            CHAR         10     Y
  EDUCLVL             SMALLINT      2     Y
  EMPNO               CHAR          6     N
  FIRSTNME            CHAR         12     Y
  HIREDATE            CHAR         10     Y
  JOBCODE             DECIMAL       3     Y
  LASTNAME            CHAR         15     Y
  MIDINIT             CHAR          1     Y
  PHONENO             CHAR          4     Y
  SALARY              DECIMAL       9     Y
  SEX                 CHAR          1     Y
  WORKDEPT            CHAR          3     Y

01/15/95                                                              PAGE 5
              VISION:BUILDER/DB2 GLOSSARY UTILITY - 2.0
                        GLOBAL DB2 TABLES GLOSSARY
          ALL TABLES FROM: INM                TO: INN
             TABLE:  INMJOBS              CREATED BY   ISPBJM1
-------------------------------------------------------------------------------
       COLUMN         COLUMN   COLUMN    NULLS          COLUMN
        NAME           TYPE    LENGTH   ALLOWED?      DESCRIPTION
-------------------------------------------------------------------------------
  BEGDATE             DATE          4     N
  COMPANY             CHAR         20     N
  EMPNO               CHAR          6     N
  ENDDATE             DATE          4     N

01/15/95                                                              PAGE 6
              ANWSWER:BUILDER/DB2 GLOSSARY UTILITY - 2.0
                        GLOBAL DB2 TABLES GLOSSARY
          ALL TABLES FROM: INM                TO: INN
             TABLE:  INMSTMP              CREATED BY   ISPBJM1
-------------------------------------------------------------------------------
       COLUMN         COLUMN   COLUMN    NULLS          COLUMN
        NAME           TYPE    LENGTH   ALLOWED?      DESCRIPTION
-------------------------------------------------------------------------------
  BEGDATE             TIMESTMP     10     N
  EMPNO               CHAR          6     N
  ENDDATE             TIMESTMP     10     N

01/15/95                                                              PAGE 7
              VISION:BUILDER/DB2 GLOSSARY UTILITY - 2.0
                        GLOBAL DB2 TABLES GLOSSARY
          ALL TABLES FROM: INM                TO: INN
             TABLE:  INMULTKY             CREATED BY   ISPBJM1
-------------------------------------------------------------------------------
       COLUMN         COLUMN   COLUMN    NULLS          COLUMN
        NAME           TYPE    LENGTH   ALLOWED?      DESCRIPTION
-------------------------------------------------------------------------------
  BRANCH              INTEGER       4     Y
  CITY                CHAR         15     Y
  DEPT                DECIMAL       9     Y
  DIV                 CHAR          3     Y

** MK4FT03  TYPE 0  END OF REPORT.

** MK4W204  TYPE 0  NUMBER OF MESSAGES PRINTED IS    4.
                    M4REPO   -       63 RECORDS INPUT
                    M4LIST   -      109 RECORDS OUTPUT
** MK4W209  TYPE 0   635168 BYTES OF MAIN STORAGE UNUSED DURING REPORTING PHASE
```

Figure A-2 Sample 1: Application Code (Page 4 of 4)

The second application is similar to the first application, except that the user supplies the table name range and, optionally, the table creator ID using the old master file. The DB2 catalog tables are accessed as user-read coordinated files with appropriate WH statements to perform the specified selection. Two different glossary format headings are printed, depending upon whether a creator ID was specified or not. The sample application is preceded by the file definitions for the application.

```
RUN     STMT    FILE    O N S D U T R A B S D S L  RB  R S M R S            START           END             SEQUENCE   LIST
  NAME    TYPE    NAME    L E E I P R E U U R L C S  PL  J O O F S              SEARCH          SEARCH           73 TO 80    SEQ
                          D W Q R D N P D F T M N T  TK  T P P O R
-------------------------------------------------------------------------------------------------------------------------
(DB2SAMP2) (RC)                                                                                                        1
** CLSAA00  TYPE 0  DEFINITION DB2SELCT IS BEING PROCESSED.                                                          *****
-------------------------------------------------------------------------------------------------------------------------
   FILE    STMT    FILE    DELETE GLOSSARY RECORD RECORD RECORDS BUFFER MAPPED      MAPPING         MAPPED          SEQUENCE   LIST
   NAME    TYPE    IDENT    FILE   LEVEL   FORMAT  SIZE  PER BLK  SIZE   FILE    INIT      TERM     RECSIZ          73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------
(DB2SELCT) (FD)             (R)      (N)      (F)    (  80)                                                             2
-------------------------------------------------------------------------------------------------------------------------
   FILE    STMT  SEGMENT  DEL SGMT VRT             SUP SEG  KEY              CREATOR        EXTERNAL          SEQUENCE   LIST
   NAME    TYPE   NAME    ETE NMBR SEG             DPL KEY CHAR               ID          TABLE NAME          73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------
(DB2SELCT) (LS) (SELPARM )    ( 1)                                                                                     3
-------------------------------------------------------------------------------------------------------------------------
   FILE    STMT   FIELD   DEL SGMT LEV FIELD  FIELD FLD SEG FLD DEC CNT  OCCRS OUTPT EDIT   DATA BASE NAME/          SEQUENCE   LIST
   NAME    TYPE   NAME    ETE NMBR NUM LOCATN LNGTH TYP KEY RND PLC SGMT NTIME EDIT  LNGH   COLUMN HEADING           73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------
(DB2SELCT) (L0) (FROM    )    ( 1) (1) (   1) ( 18) (C) (1)                                                            4
(DB2SELCT) (L0) (TO      )    ( 1) (1) (  19) ( 18) (C)                                                                5
(DB2SELCT) (L0) (CREATOR )    ( 1) (1) (  37) (  8) (C)                                                                6
** CLSDC02  TYPE 0  THE DEFINITION IS VALID AND HAS BEEN CATALOGED.                                                  *****
** CLSAA00  TYPE 0  DEFINITION SCOLUMNS IS BEING PROCESSED.                                                          *****
-------------------------------------------------------------------------------------------------------------------------
   FILE    STMT    FILE    DELETE GLOSSARY RECORD RECORD RECORDS BUFFER MAPPED      MAPPING         MAPPED          SEQUENCE   LIST
   NAME    TYPE    IDENT    FILE   LEVEL   FORMAT  SIZE  PER BLK  SIZE   FILE    INIT      TERM     RECSIZ          73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------
(SCOLUMNS) (FD)             (R)      (N)      (S)                (01024)                                                 7
COMMENT ***** ************************************************************  *****                                       8
COMMENT ***** *                                                          *   *****                                       9
COMMENT ***** *  SYSCOLUMNS TABLE                                        *   *****                                      10
COMMENT ***** *                                                          *   *****                                      11
COMMENT ***** ************************************************************  *****                                      12
-------------------------------------------------------------------------------------------------------------------------
   FILE    STMT  SEGMENT  DEL SGMT VRT             SUP SEG  KEY              CREATOR        EXTERNAL          SEQUENCE   LIST
   NAME    TYPE   NAME    ETE NMBR SEG             DPL KEY CHAR               ID          TABLE NAME          73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------
(SCOLUMNS) (LS) (SCOLUMNS)    ( 1)                                          (SYSDB2  ) (SYSCOLUMNS       )           13
-------------------------------------------------------------------------------------------------------------------------
   FILE    STMT   FIELD   DEL SGMT LEV FIELD  FIELD FLD SEG FLD DEC CNT  OCCRS OUTPT EDIT   DATA BASE NAME/          SEQUENCE   LIST
   NAME    TYPE   NAME    ETE NMBR NUM LOCATN LNGTH TYP KEY RND PLC SGMT NTIME EDIT  LNGH   COLUMN HEADING           73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------
(SCOLUMNS) (L0) (COLTYPE )    ( 1) (1) (0001) (008) (C)                                 (COLTYPE              )          14
(SCOLUMNS) (L1) (COLTYPE )                                                              (COLUMN               )          15
(SCOLUMNS) (L2) (COLTYPE )                                                              (TYPE                 )          16
(SCOLUMNS) (L0) (LENGTH  )    ( 1) (1) (0009) (007) (C)                                 (LENGTH               )          17
(SCOLUMNS) (L1) (LENGTH  )                                                              (COLUMN               )          18
(SCOLUMNS) (L2) (LENGTH  )                                                              (LENGTH               )          19
(SCOLUMNS) (L0) (TNAME   )    ( 1) (1) (0016) (018) (C) (1)                             (TENAME               )          20
(SCOLUMNS) (L1) (TNAME   )                                                              (TABLE                )          21
(SCOLUMNS) (L2) (TNAME   )                                                              (NAME                 )          22
(SCOLUMNS) (L0) (CREATOR )    ( 1) (1) (0034) (008) (C)                                 (TBCREATOR            )          23
(SCOLUMNS) (L1) (CREATOR )                                                              (TABLE                )          24
(SCOLUMNS) (L2) (CREATOR )                                                              (CREATOR              )          25
(SCOLUMNS) (L0) (NULLS   )    ( 1) (1) (0042) (001) (C)                                 (NULLS                )          26
(SCOLUMNS) (L1) (NULLS   )                                                              (NULLS                )          27
(SCOLUMNS) (L2) (NULLS   )                                                              (ALLOWED?             )          28
(SCOLUMNS) (L0) (CNAME   )    ( 1) (1)         (018) (V)                                (NAME                 )          29
(SCOLUMNS) (L1) (CNAME   )                                                              (COLUMN               )          30
(SCOLUMNS) (L2) (CNAME   )                                                              (NAME                 )          31
(SCOLUMNS) (L0) (REMARKS )    ( 1) (1)         (254) (V)                          (20) (REMARKS              )          32
(SCOLUMNS) (L1) (REMARKS )                                                              (COLUMN               )          33
(SCOLUMNS) (L2) (REMARKS )                                                              (DESCRIPTION          )          34
** CLSDC02  TYPE 0  THE DEFINITION IS VALID AND HAS BEEN CATALOGED.                                                  *****
```

Figure A-3 Sample 2: File Definitions

```
RUN     STMT    FILE    O N S D U T R A B S D S L  RB  R S M R S            START           END             SEQUENCE   LIST
   NAME    TYPE    NAME    L E E I P R E U U R L C S  PL  J O O F S              SEARCH          SEARCH           73 TO 80    SEQ
                           D W Q R D N P D F T M N T  TK  T P P O R
-------------------------------------------------------------------------------------------------------------------------------
(DB2SAMP2) (RC) (DB2SELCT) (S) (U)      (S)      (#,T)            (R)                                                        1
-------------------------------------------------------------------------------------------------------------------------------
   RUN     STMT  PARAMETER                                                                                          SEQUENCE   LIST
   NAME    TYPE    NAME                                PARAMETER VALUE                                               73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(DB2SAMP2) (RP) (DB2     )        (DB2T    INM4CALL                                    )                                    3
-------------------------------------------------------------------------------------------------------------------------------
   RUN     STMT  LOGICAL  LBL  DTF/DD      KEY       KEY       KEY      QUALIFIER   USER I/O  ARRAY       PASSWORD/  SEQUENCE   LIST
   NAME    TYPE  FILENAME       NAME     FIELD-1   FIELD-2   FIELD-3   ICF CRD DBI   MODULE   QUAL       AUTH.-ID   73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(DB2SAMP2) (RF) (SCOLUMNS) (S) (M4CORD1 )                                    (R)                                            4
(DB2SAMP2) (RF) (SCOLUMNS) (S) (M4CORD2 )                                    (R)                                            5
** MK4QK01  TYPE 1  ASCENDING SEQUENCE ASSUMED ON SEQUENTIALLY READ DB2 COORDINATED FILE SCOLUMNS                          *****
** MK4QK01  TYPE 1  ASCENDING SEQUENCE ASSUMED ON SEQUENTIALLY READ DB2 COORDINATED FILE SCOLUMNS                          *****
-------------------------------------------------------------------------------------------------------------------------------
   RUN     STMT   SEGMENT    FILE    WHERE                                                                            SEQUENCE   LIST
   NAME    TYPE    NAME      QUAL    PREDICATE                                                                         73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(DB2SAMP2) (WH)   (SCOLUMNS)  (1)    (TBNAME <= :N.TO AND TBCREATOR = :N.CREATOR      )                                  6
(DB2SAMP2) (WH)   (SCOLUMNS)  (2)    (TBNAME <= :N.TO                                 )                                  7

                                      ****************************
                                      *  PROC NAME - MAIN        *
                                      *  INPUT STREAM PROCEDURE  *
                                      ****************************
-------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                              MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID        ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM       73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(ER)                                                                                (N)                                       8
    ***** ASL Procedure Follows - Using Translator Version x.xx
    ##PROC                                                                                                    (        )    9
    ;EACH MASTER-FILE-RECORD CONTROL PROCEDURE                                                                 (        )   10
    ;                                                                                                         (        )   11
    ;  TEMPORARY FIELDS                                                                                       (        )   12
    ;                                                                                                         (        )   13
    BLANK:    FIELD C 1                                                                                       (        )   14
    TNAMEX:   FIELD C 6  INIT 'TABLE:'                                                                        (        )   15
    TABLES1:  FIELD C 8  INIT 'TABLE(S)'                                                                      (        )   16
    TABLES2:  FIELD C 10 INIT 'ALL TABLES'                                                                    (        )   17
    CREATED:  FIELD C 11 INIT 'CREATED BY'                                                                    (        )   18
    TOX:      FIELD C 3  INIT 'TO:'                                                                           (        )   19
    FROMX:    FIELD C 5  INIT 'FROM:'                                                                         (        )   20
    ;                                                                                                         (        )   21
    ;  CONTROL STATEMENTS PROCESSING                                                                          (        )   22
    ;                                                                                                         (        )   23
    IF PF(N.CREATOR 1 1) NE ' '                  ;CREATOR ID SPECIFIED?                                       (        )   24
       IF FIND(1.SCOLUMNS WHERE 1.TNAME GE N.FROM) ;REPOSITION CURSOR AND                                     (        )   25
                                                 ;SELECT BY CREATOR+TABLE                                    (        )   26
          DO UNTIL NOT FIND(1.SCOLUMNS)                                                                       (        )   27
             CALL REPORT LISTCRE8                ;REPORT BY CREATOR+TABLE                                     (        )   28
          END                                                                                                 (        )   29
       END                                                                                                    (        )   30
    ELSE                                                                                                      (        )   31
       IF FIND(2.SCOLUMNS WHERE 2.TNAME GE N.FROM) ;REPOSITION CURSOR AND                                     (        )   32
          DO UNTIL NOT FIND(2.SCOLUMNS)            ;SELECT BY TABLE ONLY                                     (        )   33
             CALL REPORT LISTNAME                  ;REPORT BY TABLE ONLY                                     (        )   34
          END                                                                                                 (        )   35
       END                                                                                                    (        )   36
    END                                                                                                       (        )   37
    ##PEND                                                                                                    (        )   38
    ***** End of ASL Procedure

                                      ****************************
                                      *  REQUEST NAME - LISTCRE8 *
                                      *  INPUT STREAM REQUEST    *
                                      ****************************
-------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                              MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID        ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM       73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(ER) (TODAY )                                                                       (S)                                       39
COMMENT ***** DB2 TABLES GLOSSARY BY CREATOR ID                                *****                                        40
-------------------------------------------------------------------------------------------------------------------------------
 ST SUM V  8  PG  PG   I SP  MAXIMUM TB COLUMN  DT  PG  STRT LIN SL RH .......SUBFILE........ C E A S R S N      C  SEQUENCE   LIST
 TY RPT S LPI WID HGT  M FRM LPP PGS TO HEADING POS POS PAGE NUM UB PN ENT    NAME   BLK/ FMT H F G T S U D      O  73 TO 80    SEQ
 MP     P             G            LT TYP POS         NUM      ML TD REC          BUF      C C S L T M S      N
-------------------------------------------------------------------------------------------------------------------------------
```

Figure A-4 Sample 2: Application Code (Page 1 of 4)

```
(E1)         ( 80)                                                                                                              41
----------------------------------------------------------------------------------------------------------------------------------
         SP  Q                E  N  S  D  C  S  T  C  C  M  M  A  G  D Q PCT/RAT     P                  PARTIAL  SEQUENCE  LIST
STMT SEQ  BF  L  FIELD   SEG-  N  P  E  E  T  U  0  U  N  A  I  V  M  E L  FIELD   RA- C      PICTURE    STRT NO  73 TO 80   SEQ
TYPE NO . COL F  NAME     LVL  D  R  Q  S  L  B  T  M  T  X  N  G  R  C F  NAME     TIO T      EDIT       CHR CHR
----------------------------------------------------------------------------------------------------------------------------------
(R1)         (T,TABLES1 )                (1) (P)                                                                                42
(R1)     ( 1)(T,FROMX   )                (1) (P)                                                                                43
(R1)     ( 1)(N,FROM    ) 01-1           (1) (P)                                                                                44
(R1)     ( 1)(T,TOX     )                (1) (P)                                                                                45
(R1)     ( 1)(N,TO      ) 01-1           (1) (P)                                                                                46
(R1)     ( 1)(T,CREATED )                (1) (P)                                                                                47
(R1)     ( 1)(1,CREATOR ) 01-1 (Y)       (1) (P)                                                                                48
(R1)         (T,BLANK   )      (Y)       (1) (P)                                                                                49
(R1)     (27)(T,TNAMEX  )                (1) (P)                                                                                50
(R1)         (1,TNAME   ) 01-1      (1)  (1) (P)                                                                                51
(R1)         (1,CNAME   ) 01-1                                                                                                  52
(R1)         (1,COLTYPE ) 01-1                                                                                                  53
(R1)         (1,LENGTH  ) 01-1                                                                                                  54
(R1)         (1,NULLS   ) 01-1                                                                                                  55
(R1)         (1,REMARKS ) 01-1                                                                                                  56
----------------------------------------------------------------------------------------------------------------------------------
STMT  SEQ                                                                                                      SEQUENCE  LIST
TYPE  NO.                                          TEXT                                                        73 TO 80  SEQ
----------------------------------------------------------------------------------------------------------------------------------
(T1)                    (                     MARK IV / DB2 GLOSSARY UTILITY - 2.0  )                                           57
(T1)                    (                     #                                     )                                           58
(T1)                                                                                                                            59
(T1)                    (                     #                                     )                                           60
(T1)                    (                       DB2 TABLES GLOSSARY BY CREATOR ID   )                                           61
(T1)                    (                     #                                     )                                           62

                                               *****************************
                                               *  REQUEST NAME - LISTNAME  *
                                               *  INPUT STREAM REQUEST     *
                                               *****************************
----------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                          MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ         BACK BR RNT AB       SEQUENCE  LIST
TYPE  DATE          REQUESTOR ID    ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM       73 TO 80   SEQ
----------------------------------------------------------------------------------------------------------------------------------
(ER)(TODAY )                                                           (S)                                                      63
COMMENT ***** DB2 TABLES GLOSSARY                                  *****                                                        64
----------------------------------------------------------------------------------------------------------------------------------
 ST SUM V  8  PG  PG   I SP  MAXIMUM TB COLUMN  DT  PG  STRT LIN SL RH .......SUBFILE........ C E A S R S N      C  SEQUENCE  LIST
 TY RPT S LPI WID HGT  M FRM LPP PGS TO HEADING POS POS PAGE NUM UB PN ENT     NAME   BLK/ FMT H F G T S U D      O  73 TO 80   SEQ
 MP     P              G             LT TYP POS         NUM      ML TD REC            BUF     C C S L T M S      N
----------------------------------------------------------------------------------------------------------------------------------
(E1)         ( 80)                                                                                                              65
----------------------------------------------------------------------------------------------------------------------------------
         SP  Q                E  N  S  D  C  S  T  C  C  M  M  A  G  D Q PCT/RAT     P                  PARTIAL  SEQUENCE  LIST
STMT SEQ  BF  L  FIELD   SEG-  N  P  E  E  T  U  0  U  N  A  I  V  M  E L  FIELD   RA- C      PICTURE    STRT NO  73 TO 80   SEQ
TYPE NO.  COL F  NAME     LVL  D  R  Q  S  L  B  T  M  T  X  N  G  R  C F  NAME     TIO T      EDIT       CHR CHR
----------------------------------------------------------------------------------------------------------------------------------
(R1)     (11)(T,TABLES2 )                (1) (P)                                                                                66
(R1)     ( 1)(T,FROMX   )                (1) (P)                                                                                67
(R1)     ( 1)(N,FROM    ) 01-1           (1) (P)                                                                                68
(R1)     ( 1)(T,TOX     )                (1) (P)                                                                                69
(R1)     ( 1)(N,TO      ) 01-1 (Y)       (1) (P)                                                                                70
(R1)         (T,BLANK   )      (Y)       (1) (P)                                                                                71
(R1)     (15)(T,TNAMEX  )                (1) (P)                                                                                72
(R1)         (2,TNAME   ) 01-1      (2)  (1) (P)                                                                                73
(R1)         (T,CREATED )                (1) (P)                                                                                74
(R1)         (2,CREATOR ) 01-1      (1)  (1) (P)                                                                                75
(R1)         (2,CNAME   ) 01-1                                                                                                  76
(R1)         (2,COLTYPE ) 01-1                                                                                                  77
(R1)         (2,LENGTH  ) 01-1                                                                                                  78
(R1)         (2,NULLS   ) 01-1                                                                                                  79
(R1)         (2,REMARKS ) 01-1                                                                                                  80
----------------------------------------------------------------------------------------------------------------------------------
STMT  SEQ                                                                                                      SEQUENCE  LIST
TYPE  NO.                                          TEXT                                                        73 TO 80  SEQ
----------------------------------------------------------------------------------------------------------------------------------
(T1)                    (                     MARK IV / DB2 GLOSSARY UTILITY - 2.0  )                                           81
(T1)                    (                     #                                     )                                           82
(T1)                                                                                                                            83
(T1)                    (                     #                                     )                                           84
(T1)                    (                              DB2 TABLES GLOSSARY          )                                           85
(T1)                    (                     #                                     )                                           86

                                    *******************************************
                                    *   SUMMARY OF GENERATED DB2 STATEMENTS   *
                                    *******************************************
----------------------------------------------------------------------------------------------------------------------------------
FILE>SEGMENT                  DB2 STATEMENT                                                 COMMENTS
```

Figure A-4        Sample 2: Application Code (Page 2 of 4)

```
-----------------------------------------------------------------------------------------------------------------------------
M4CORD1>SCOLUMNS   STATEMENT  1
          SELECT COLTYPE,LENGTH,TBNAME,TBCREATOR,NULLS,NAME,REMARKS        SELECT USING KEY (GREATER THAN OR EQUAL) OF ROOT
          FROM SYSDB2.SYSCOLUMNS X1 WHERE (TBNAME>=?) AND TBNAME <= ?      SEGMENT. WHERE CLAUSE BUILT FROM: RD OPERATOR, AND
          AND TBCREATOR = ? ORDER BY TBNAME                                (WH) STATEMENT. DB2 CALL(S): 1 PREPARE(S), 1 OPEN(S),
          HOST VARIABLE(S) = N.TO N.CREATOR                                AND 18 FETCH(ES).
M4CORD2>SCOLUMNS   STATEMENT  2
          SELECT COLTYPE,LENGTH,TBNAME,TBCREATOR,NULLS,NAME,REMARKS        SELECT USING KEY (GREATER THAN OR EQUAL) OF ROOT
          FROM SYSDB2.SYSCOLUMNS X1 WHERE (TBNAME>=?) AND TBNAME <= ?      SEGMENT. WHERE CLAUSE BUILT FROM: RD OPERATOR, AND
          ORDER BY TBNAME                                                  (WH) STATEMENT. DB2 CALL(S): 1 PREPARE(S), 1 OPEN(S),
          HOST VARIABLE(S) = N.TO                                          AND 29 FETCH(ES).

** MK4W204  TYPE 0  NUMBER OF MESSAGES PRINTED IS    2.
                   M4OLD    -        2 RECORDS INPUT
                   M4CORD1  -       18 RECORDS INPUT
                   M4CORD2  -       29 RECORDS INPUT
                   M4REPO   -       80 RECORDS OUTPUT
                   M4INPUT  -       86 RECORDS INPUT
                   M4LIST   -      251 RECORDS OUTPUT
                      3 TRACKS ASSIGNED TO M4LIB    --    2 TRACKS NOT FULL. LIBRARY DIRECTORY BLOCKING FACTOR IS  138.
** MK4W209  TYPE 0  1016328 BYTES OF MAIN STORAGE UNUSED DURING DECODING PHASE
                   1019080 BYTES OF MAIN STORAGE UNUSED DURING COMPILATION PHASE
                    634880 BYTES OF MAIN STORAGE UNUSED DURING PROCESSING PHASE** MK4ED02  TYPE 0  REQUEST-LISTCRE8  REPORT-1  REQU
ESTOR ID-

01/15/95                                                                  PAGE 1
               VISION:BUILDER/DB2 GLOSSARY UTILITY - 2.0
                      DB2 TABLES GLOSSARY BY CREATOR ID
  TABLE(S) FROM: DEPA              TO: DEPZ              CREATED BY  ISPBJM1
                          TABLE:  DEPTEMPL
-------------------------------------------------------------------------------
        COLUMN         COLUMN   COLUMN    NULLS          COLUMN
         NAME           TYPE    LENGTH   ALLOWED?      DESCRIPTION
-------------------------------------------------------------------------------
  DEPTNAME            CHAR           36     Y
  DEPTNO              CHAR            3     Y
  EMPNO               CHAR            6     Y
  FIRSTNME            CHAR           12     Y
  JOBCODE             DECIMAL         3     Y
  LASTNAME            CHAR           15     Y
  MIDINIT             CHAR            1     Y
  PHONENO             CHAR            4     Y
  SALARY              DECIMAL         9     Y

01/15/95                                                                  PAGE 2
               VISION:BUILDER/DB2 GLOSSARY UTILITY - 2.0
                      DB2 TABLES GLOSSARY BY CREATOR ID
  TABLE(S) FROM: DEPA              TO: DEPZ              CREATED BY  ISPBJM1
                          TABLE:  DEPT118G
-------------------------------------------------------------------------------
        COLUMN         COLUMN   COLUMN    NULLS          COLUMN
         NAME           TYPE    LENGTH   ALLOWED?      DESCRIPTION
-------------------------------------------------------------------------------
   DEPTNAME            CHAR           36     Y
   DEPTNO              CHAR            3     Y
   EMPNO               CHAR            6     Y
   SALARY              DECIMAL         9     Y
   TEMP1               CHAR            1     Y
   TEMP2               DECIMAL         1     Y
   TEMP3               DECIMAL         1     Y
   TEMP4               SMALLINT        2     Y
   TEMP5               FLOAT           8     Y

** MK4FT03  TYPE 0  END OF REPORT.

** MK4ED02  TYPE 0  REQUEST-LISTNAME  REPORT-1  REQUESTOR ID-

01/15/95                                                                  PAGE 1
               VISION:BUILDER/DB2 GLOSSARY UTILITY - 2.0
                             DB2 TABLES GLOSSARY
          ALL TABLES FROM: INTD              TO: INTF
              TABLE:  INTDEPT            CREATED BY   INF1
-------------------------------------------------------------------------------
        COLUMN         COLUMN   COLUMN    NULLS          COLUMN
         NAME           TYPE    LENGTH   ALLOWED?      DESCRIPTION
-------------------------------------------------------------------------------
  MGRNO               CHAR            6     N
  ADMRDEPT            CHAR            3     N
  DEPTNAME            VARCHAR        36     N
  DEPTNO              CHAR            3     N

01/15/95                                                                  PAGE 2
               VISION:BUILDER/DB2 GLOSSARY UTILITY - 2.0
                             DB2 TABLES GLOSSARY
```

Figure A-4 Sample 2: Application Code (Page 3 of 4)

```
          ALL TABLES FROM: INTD              TO: INTF
              TABLE:  INTDSPTXT           CREATED BY   INF1
-------------------------------------------------------------------------------
       COLUMN         COLUMN   COLUMN    NULLS          COLUMN
        NAME           TYPE    LENGTH   ALLOWED?     DESCRIPTION
-------------------------------------------------------------------------------
  DSPINDEX           CHAR           2     N
  DSPLINE            CHAR          79     N
  LINENO             CHAR           2     N

01/15/95                                                              PAGE 3
                VISION:BUILDER/DB2 GLOSSARY UTILITY - 2.0
                             DB2 TABLES GLOSSARY
          ALL TABLES FROM: INTD              TO: INTF
              TABLE:  INTEMPL             CREATED BY   INF1
-------------------------------------------------------------------------------
       COLUMN         COLUMN   COLUMN    NULLS          COLUMN
        NAME           TYPE    LENGTH   ALLOWED?     DESCRIPTION
-------------------------------------------------------------------------------
  BRTHDATE           DATE           4     Y
  EDUCLVL            SMALLINT       2     Y
  EMPNO              CHAR           6     N
  FIRSTNME           VARCHAR       12     N
  HIREDATE           DATE           4     Y
  JOBCODE            DECIMAL        3     Y
  LASTNAME           VARCHAR       15     N
  MIDINIT            CHAR           1     N
  PHONENO            CHAR           4     Y
  SALARY             DECIMAL        8     Y
  SEX                CHAR           1     Y
  WORKDEPT           CHAR           3     N

01/15/95                                                              PAGE 4
                VISION:BUILDER/DB2 GLOSSARY UTILITY - 2.0
                             DB2 TABLES GLOSSARY
          ALL TABLES FROM: INTD              TO: INTF
              TABLE:  INTEMPRAC           CREATED BY   INF1
-------------------------------------------------------------------------------
       COLUMN         COLUMN   COLUMN    NULLS          COLUMN
        NAME           TYPE    LENGTH   ALLOWED?     DESCRIPTION
-------------------------------------------------------------------------------
  EMSTDATE           DATE           4     Y
  PROJNO             CHAR           6     N
  ACTNO              SMALLINT       2     N
  EMENDATE           DATE           4     Y
  EMPNO              CHAR           6     N
  EMPTIME            DECIMAL        5     Y

01/15/95                                                              PAGE 5
                VISION:BUILDER/DB2 GLOSSARY UTILITY - 2.0
                             DB2 TABLES GLOSSARY
          ALL TABLES FROM: INTD              TO: INTF
              TABLE:  INTDEPT             CREATED BY   ISPIPG2
-------------------------------------------------------------------------------
       COLUMN         COLUMN   COLUMN    NULLS          COLUMN
        NAME           TYPE    LENGTH   ALLOWED?     DESCRIPTION
-------------------------------------------------------------------------------
  ADMRDEPT           CHAR           3     Y
  DEPTNAME           CHAR          36     N
  DEPTNO             CHAR           3     Y
  MGRNO              CHAR           6     Y

** MK4FT03  TYPE 0  END OF REPORT.

** MK4MW204  TYPE 0  NUMBER OF MESSAGES PRINTED IS    4.
                    M4REPO   -       80 RECORDS INPUT
                    M4LIST   -      116 RECORDS OUTPUT
** MK4MW209  TYPE 0   647456 BYTES OF MAIN STORAGE UNUSED DURING REPORTING PHASE
```

Figure A-4 Sample 2: Application Code (Page 4 of 4)

Chapter

# B VISION:Builder System Component Relationships

Figure B-1 illustrates the relationship of the various components of VISION:Builder to each other. The components of VISION:Builder are the base VISION:Builder System, COMLIB, VISION:Workbench for DOS, and VISION:Workbench for ISPF.

VISION:Builder applications can be prepared using either VISION:Workbench for DOS, VISION:Workbench for ISPF, or your favorite TSO editor. In all cases, ultimately the VISION:Builder source program is stored in a source library from which it is compiled and executed.

When using VISION:Workbench for ISPF or your favorite TSO editor, your program is saved directly into the source library using the edit process. When VISION:Workbench for DOS is used, the application code must be exported from VISION:Workbench for DOS and transferred from your workstation to the source library on your host computer. This transfer process can be accomplished with whatever file transfer facility you have at your site. VISION:Workbench for DOS itself does not initiate or perform the process nor does it impose any requirements for a specific file transfer protocol.

VISION:Workbench for ISPF interacts directly with VISION:Builder for the purpose of syntax validation and with COMLIB for the purpose of storing and retrieving definitions from the common library.

Applications resident in the source library can be edited and modified with any of the tools as shown in Figure B-1. For example, an application can be imported to VISION:Workbench for DOS for modification. VISION:Workbench for DOS gives you the advantage of full syntax validation and report format simulation. The application can be exported for testing. Additional code for features not currently supported by VISION:Workbench for DOS (such as WH statements and miscellaneous run-time parameters), final tuning, and correction can be performed using VISION:Workbench for ISPF or TSO as necessary.

Likewise, new applications can be developed using VISION:Workbench for DOS and exported for testing and final source maintenance by VISION:Workbench for ISPF or TSO. The use of a central source library allows you to take advantage of the appropriate source development and maintenance tool for each particular situation.

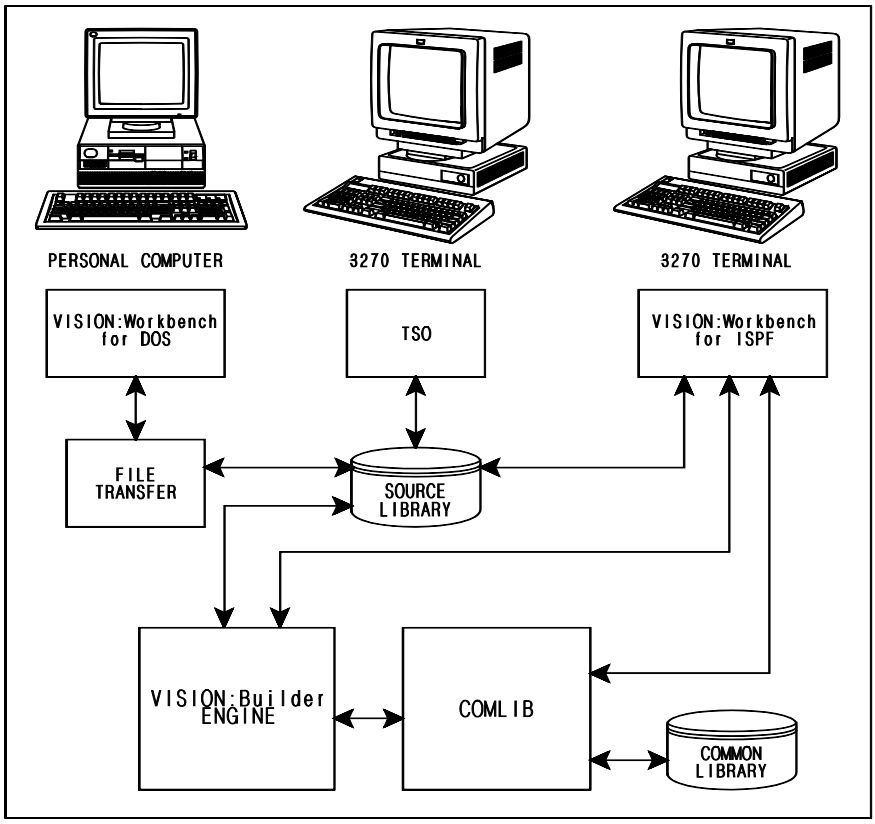



Figure B-1 VISION:Builder System Component Relationships

Chapter

C

# DB2 Reserved Words

Figure C-1 contains a list of DB2 reserved words. While VISION:Builder does not have reserved words, special consideration must be made if any of the words in this list are used as field names or as DB2 external names.

If a DB2 reserved word is used as a DB2 external name (such as a table name or a column name) in the file definition, COMLIB catalogs it as long as it is syntactically correct. However, during VISION:Builder execution, DB2 abends due to the improper use of a reserved word in the DB2 statement. You can enter a reserved word as a delimited identifier in the DB2 external name entry (by enclosing the entry between double quotation marks), allowing the VISION:Builder run to complete successfully.

If a DB2 reserved word is used as a VISION:Builder field name and the DB2 external name is blank, DB2 abends if that field is output to a VISION:Builder subfile with a format of S. This occurs because VISION:Builder uses the VISION:Builder field name as the DB2 external name if the external name field is left blank; therefore, DB2 is passed a reserved word as a field name and may not accept it.

| | | | | |
|---|---|---|---|---|
| ADD | CURSOR | GOTO | OF | SYNONYM |
| ALL | DATABASE | GRANT | ON | TABLE |
| ALTER | DELETE | GROUP | OPTIMIZE | TABLESPACE |
| AND | DESCRIPTOR | HAVING | OR | TO |
| ANY | DISTINCT | IMMEDIATE | ORDER | UNION |
| AS | DROP | IN | PACKAGE | UNIQUE |
| AUDIT | EDITPROC | INDEX | PART | UPDATE |
| BETWEEN | END-EXEC | INSERT | PLAN | USER |
| BUFFERPOOL | ERASE | INTO | PRIQTY | USING |
| BY | ESCAPE | IS | PRIVILEGES | VALIDPROC |
| CLUSTER | EXCEPT | KEY | PROGRAM | VALUES |
| COLLECTION | EXECUTE | LIKE | REVOKE | VCAT |
| COLUMN | EXISTS | LOCKSIZE | SECQTY | VIEW |
| CONCAT | FIELDPROC | NOT | SELECT | VOLUMES |
| COUNT | FOR | NULL | SET | WHERE |
| CREATE | FROM | NUMPARTS | SOME | WITH |
| CURRENT | GO | OBID | STOGROUP | |

Figure C-1 DB2 Reserved Words

Appendix

# D Complete Sample Applications

This appendix contains the complete versions of sample applications that have been described previously in this guide.


- The Application Cycle on page D-1
- Input File Processing on page D-5
- Output File Processing on page D-60
- Data Structure Handling on page D-65
- Reporting on page D-77
- Arrays on page D-90
- Extended Reporting Capabilities on page D-127
- Advanced Transaction Processing on page D-137
- Report Manager on page D-179
- Cataloging on page D-211


## The Application Cycle

### Sample Application

This is the complete version of Figure 5-17 on page 5-18.

```
-------------------------------------------------------------------------------------------------------------------------------
   RUN     STMT    FILE    O N S D U T R A B S D S L  RB  R S M R S          START          END            SEQUENCE   LIST
   NAME    TYPE    NAME    L E E I P R E U U R L C S  PL  J O O F S          SEARCH         SEARCH         73 TO 80    SEQ
                           D W Q R D N P D F T M N T  TK  T P P O R
-------------------------------------------------------------------------------------------------------------------------------
 (ASLRC    ) (RC) (TEMPL    ) (S) (U)      (S)          (G)          (R)                                  (ASLGEN'D)      1
0     CONTROL DB2 D61A INM4CALL   LISTGEN
      FILE MASTER INPUT, NAME TEMPL, KEYS NONE
      FILE REPORT
      FILE SUBF3, NAME OUT1
      ;
-------------------------------------------------------------------------------------------------------------------------------
   RUN     STMT  LOGICAL  LBL  DTF/DD      KEY       KEY       KEY      QUALIFIER   USER I/O  ARRAY       PASSWORD/  SEQUENCE   LIST
```

Figure D-1 Sample Application (Page 1 of 4)

```
   NAME    TYPE  FILENAME      NAME     FIELD-1   FIELD-2   FIELD-3   ICF CRD DBI   MODULE   QUAL       AUTH.-ID  73 TO 80    SEQ
-----------------------------------------------------------------------------------------------------------------------------------
(ASLRC   ) (RF) (TEMPL   ) (S) (M4OLD   )                                                                          (ASLGEN'D)     2
(ASLRC   ) (RF) (OUT1    ) (S) (M4SUBF3 )                                                                          (ASLGEN'D)     3
-----------------------------------------------------------------------------------------------------------------------------------
   RUN     STMT  PARAMETER                                                                                          SEQUENCE   LIST
   NAME    TYPE    NAME                          PARAMETER VALUE                                                    73 TO 80    SEQ
-----------------------------------------------------------------------------------------------------------------------------------
(ASLRC   ) (RP) (DB2     )        (D61A    INM4CALL                                 )                              (ASLGEN'D)     4


=================================================================================================================================
=                                                                                                                               =
=                                            F I L E   S U M M A R Y                                                            =
=                                                                                                                               =
= JOBNAME: HANME025                         STEPNAME: M4RUN                                                                     =
=                                                                                                                               =
=  FILE NAME: M4OLD                         TABLE:  DSN8610.EMP                                                                 =
=  FILE NAME: M4SUBF3                       DSNAME: HANME02.VARIABLE                                                            =
=  FILE NAME: M4LIB                         DSNAME: HANME02.COMLIB14                                                            =
=                                                                                                                               =
=================================================================================================================================
1NOV 05, 2001  12.24.52                                                                                            PAGE      3
                                                  ****************************
                                                  *  PROC NAME - MAIN        *
                                                  *  INPUT STREAM PROCEDURE  *
                                                  ****************************
-----------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                              MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID        ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM      73 TO 80    SEQ
-----------------------------------------------------------------------------------------------------------------------------------
 (ER)                                                                          (N)                                 (ASLGEN'D)     5
0     MAIN: PROC
      ; Each master file record control procedure
      COUNTER: FIELD F 4 INIT 0
      ;
      IF N.JOB = 'PRES'
         LET T.TEMP1 = 'REPORT ONLY IF JOB IS PRES'
         LET T.COUNTER = T.COUNTER + 1
         IF T.COUNTER GT 5
            LET PF(F.EOF 1 1) = 'E'
         ELSE
            CALL REPORT REPORT1
            CALL REPORT REPORT2
            CALL SUBFILE SUBFILE3
         END
      END
      END PROC
      ;
-----------------------------------------------------------------------------------------------------------------------------------
STMT   FIELD    FIELD FLD DEC OUTPT EDIT      INITIAL         LINE 1 OF       LINE 2 OF                              SEQUENCE   LIST
TYPE   NAME     LNGTH TYP PLC EDIT  LGTH       VALUE        COLUMN HEADING  COLUMN HEADING                         73 TO 80    SEQ
-----------------------------------------------------------------------------------------------------------------------------------
(TF) (COUNTER ) (  4) (F)               (0              )                                                          (ASLGEN'D)     6
-----------------------------------------------------------------------------------------------------------------------------------
STMT SEQ  LOG CON .....OPERAND-A.....  OPER  .............OPERAND-B.............   ......RESULT......  PARTIAL FIELD  SEQUENCE   LIST
TYPE NO.  LEV CTR QLF FIELD   SEG-LVL  AIN   QLF FIELD   SEG-LVL  (OR CONSTANT)   QLF FIELD  SEG-LVL  STRT NUMB OPR  73 TO 80    SEQ
-----------------------------------------------------------------------------------------------------------------------------------
(PR) (100)          (N,JOB     ) 001-1  (EQ)    (C,PRES                         )                                   (ASLGEN'D)     7
(PR) (101)                           (NS)      (END                            )                                   (ASLGEN'D)     8
(PR) (102)                           (R )    (C,REPORT ONLY IF JOB IS PRES     ) (T,TEMP1    )                     (ASLGEN'D)     9
** MK4DI02  TYPE 0  DEFINITION OF TEMPORARY FIELD TEMP1    - FORMAT C  LENGTH  26  DECIMALS 0 EDIT LENGTH  26 .                 *****
(PR) (103)          (T,COUNTER )         (+ )    (D,1                            ) (T,COUNTER )                      (ASLGEN'D)    10
(PR) (104)          (T,COUNTER )         (GT)    (D,5                            )                                   (ASLGEN'D)    11
(PR) (105)                           (NS)      (108                            )                                   (ASLGEN'D)    12
(PR) (106)                           (R )    (C,E                              ) (F,EOF      )         ( 1) ( 1) (C) (ASLGEN'D)    13
(PR) (107)                           (GO)      (END                            )                                   (ASLGEN'D)    14
(PR) (108)                           (GO)      (SUB REPORT1                    )                                   (ASLGEN'D)    15
(PR) (109)                           (GO)      (SUB REPORT2                    )                                   (ASLGEN'D)    16
(PR) (110)                           (GO)      (SUB SUBFILE3                   )                                   (ASLGEN'D)    17
1NOV 05, 2001  12.24.52                                                                                            PAGE      4
                                                  ****************************
                                                  *  PROC NAME - ENDOFILE    *
                                                  *  INPUT STREAM PROCEDURE  *
                                                  ****************************
-----------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                              MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID        ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM      73 TO 80    SEQ
-----------------------------------------------------------------------------------------------------------------------------------
 (ER)                                                                          (E)                                 (ASLGEN'D)    18
0     ENDOFILE: PROC TYPE EOF
      ; End of file control procedure
      LET T.TEMP1 = 'ALL DONE'
         REPORT T.TEMP1 T.COUNTER
         END REPORT
```

Figure D-1 Sample Application (Page 2 of 4)

```
    END PROC
    ;
-------------------------------------------------------------------------------------------------------------------------------
STMT SEQ  LOG CON .....OPERAND-A.....  OPER  .............OPERAND-B.............   ......RESULT......  PARTIAL FIELD  SEQUENCE   LIST
TYPE NO.  LEV CTR QLF FIELD   SEG-LVL  ATN   QLF FIELD   SEG-LVL  (OR CONSTANT)   QLF FIELD  SEG-LVL  STRT NUMB OPR  73 TO 80   SEQ
-------------------------------------------------------------------------------------------------------------------------------
(PR) (100)                            (R )   (C,ALL DONE                      ) (T,TEMP1    )                    (ASLGEN'D)    19
(PR) (101)                            (GO)     (SUB REPT___3                  )                                  (ASLGEN'D)    20
1NOV 05, 2001  12.24.52                                                                                             PAGE     5

                                                 ****************************
                                                 *  REQUEST NAME - REPT___3 *
                                                 *  GENERATED REQUEST       *
                                                 ****************************

-------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                                MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID          ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM       73 TO 80   SEQ
-------------------------------------------------------------------------------------------------------------------------------
(ER)                                                                              (S)                              (ASLGEN'D)    21
-------------------------------------------------------------------------------------------------------------------------------
 ST SUM V  8  PG  PG   I SP  MAXIMUM TB COLUMN  DT  PG  STRT LIN SL RH .......SUBFILE........ C E A S R S N      C  SEQUENCE   LIST
 TY RPT S LPI WID HGT  M FRM LPP PGS TO HEADING POS POS PAGE NUM UB PN ENT    NAME   BLK/ FMT H F G T S U D      O  73 TO 80   SEQ
 MP     P            G            LT TYP POS         NUM      ML TD REC          BUF      C C S L T M S      N
-------------------------------------------------------------------------------------------------------------------------------
(E1)                                                                                                            (ASLGEN'D)    22
-------------------------------------------------------------------------------------------------------------------------------
         SP  Q              E N S  D C S T C C M M A G D Q PCT/RAT     P               PARTIAL   SEQUENCE   LIST
STMT SEQ BF  L  FIELD  SEG- N P E  E T U O U N A I V M E L FIELD  RA- C     PICTURE     STRT NO  73 TO 80   SEQ
TYPE NO. COL F  NAME     LVL D R Q  S L B T M T X N G R C F NAME   TIO T      EDIT      CHR CHR
-------------------------------------------------------------------------------------------------------------------------------
(R1)         (T,TEMP1    )                                                                                      (ASLGEN'D)    23
(R1)         (T,COUNTER  )                                                                                      (ASLGEN'D)    24
1NOV 05, 2001  12.24.52                                                                                             PAGE     6

                                                 ****************************
                                                 *  PROC NAME - REPORT1     *
                                                 *  INPUT STREAM PROCEDURE  *
                                                 ****************************

-------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                                MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID          ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM       73 TO 80   SEQ
-------------------------------------------------------------------------------------------------------------------------------
(ER)                                                                              (S)                              (ASLGEN'D)    25
0     REPORT1: REPORT TYPE SUBROUTINE,
               ITEMS T.TEMP1, N.EMPNO, N.FIRSTNME, N.LASTNAME
    END REPORT
    ;
-------------------------------------------------------------------------------------------------------------------------------
 ST SUM V  8  PG  PG   I SP  MAXIMUM TB COLUMN  DT  PG  STRT LIN SL RH .......SUBFILE........ C E A S R S N      C  SEQUENCE   LIST
 TY RPT S LPI WID HGT  M FRM LPP PGS TO HEADING POS POS PAGE NUM UB PN ENT    NAME   BLK/ FMT H F G T S U D      O  73 TO 80   SEQ
 MP     P            G            LT TYP POS         NUM      ML TD REC          BUF      C C S L T M S      N
-------------------------------------------------------------------------------------------------------------------------------
(E1)                                                                                                            (ASLGEN'D)    26
-------------------------------------------------------------------------------------------------------------------------------
         SP  Q              E N S  D C S T C C M M A G D Q PCT/RAT     P               PARTIAL   SEQUENCE   LIST
STMT SEQ BF  L  FIELD  SEG- N P E  E T U O U N A I V M E L FIELD  RA- C     PICTURE     STRT NO  73 TO 80   SEQ
TYPE NO. COL F  NAME     LVL D R Q  S L B T M T X N G R C F NAME   TIO T      EDIT      CHR CHR
-------------------------------------------------------------------------------------------------------------------------------
(R1)         (T,TEMP1    )                                                                                      (ASLGEN'D)    27
(R1)         (N,EMPNO    ) 001-1                                                                                (ASLGEN'D)    28
(R1)         (N,FIRSTNME) 001-1                                                                                 (ASLGEN'D)    29
(R1)         (N,LASTNAME) 001-1                                                                                 (ASLGEN'D)    30
1NOV 05, 2001  12.24.52                                                                                             PAGE     7

                                                 ****************************
                                                 *  PROC NAME - REPORT2     *
                                                 *  INPUT STREAM PROCEDURE  *
                                                 ****************************

-------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                                MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID          ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM       73 TO 80   SEQ
-------------------------------------------------------------------------------------------------------------------------------
(ER)                                                                              (S)                              (ASLGEN'D)    31
0     REPORT2: REPORT TYPE SUBROUTINE, ITEMS T.COUNTER F.EOF
    END REPORT
    ;
-------------------------------------------------------------------------------------------------------------------------------
 ST SUM V  8  PG  PG   I SP  MAXIMUM TB COLUMN  DT  PG  STRT LIN SL RH .......SUBFILE........ C E A S R S N      C  SEQUENCE   LIST
 TY RPT S LPI WID HGT  M FRM LPP PGS TO HEADING POS POS PAGE NUM UB PN ENT    NAME   BLK/ FMT H F G T S U D      O  73 TO 80   SEQ
 MP     P            G            LT TYP POS         NUM      ML TD REC          BUF      C C S L T M S      N
-------------------------------------------------------------------------------------------------------------------------------
(E1)                                                                                                            (ASLGEN'D)    32
-------------------------------------------------------------------------------------------------------------------------------
         SP  Q              E N S  D C S T C C M M A G D Q PCT/RAT     P               PARTIAL   SEQUENCE   LIST
STMT SEQ BF  L  FIELD  SEG- N P E  E T U O U N A I V M E L FIELD  RA- C     PICTURE     STRT NO  73 TO 80   SEQ
TYPE NO. COL F  NAME     LVL D R Q  S L B T M T X N G R C F NAME   TIO T      EDIT      CHR CHR
-------------------------------------------------------------------------------------------------------------------------------
(R1)         (T,COUNTER  )                                                                                      (ASLGEN'D)    33
(R1)         (F,EOF      )                                                                                      (ASLGEN'D)    34
```

Figure D-1 Sample Application (Page 3 of 4)

```
1NOV 05, 2001  12.24.52                                                                                                PAGE      8
                                                ****************************
                                                *  PROC NAME - SUBFILE3    *
                                                *  INPUT STREAM PROCEDURE  *
                                                ****************************

-------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                                MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID          ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM      73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
 (ER)                                                                                (S)                          (ASLGEN'D)    35
0     SUBFILE3: EXTRACT FILE OUT1, TYPE SUBROUTINE,
        ITEMS T.TEMP1    T.COUNTER  N.EMPNO N.JOB,
              N.LASTNAME N.FIRSTNME F.EOF
-------------------------------------------------------------------------------------------------------------------------------
 ST SUM V  8  PG  PG   I SP  MAXIMUM TB COLUMN  DT  PG  STRT LIN SL RH .......SUBFILE........ C E A S R S N      C  SEQUENCE   LIST
 TY RPT S LPI WID HGT  M FRM LPP PGS TO HEADING POS POS PAGE NUM UB PN ENT    NAME   BLK/ FMT H F G T S U D      O  73 TO 80    SEQ
 MP     P             G            LT TYP POS         NUM      ML TD REC           BUF      C C S L T M S      N
-------------------------------------------------------------------------------------------------------------------------------
 (E1)                                                                  (NR)   (OUT1    )                          (ASLGEN'D)    36
 ** MK4FR75  TYPE 0  SUBFILE RECORD FORMAT IS DEFAULTED TO A V-TYPE.                                                  *****
-------------------------------------------------------------------------------------------------------------------------------
          SP  Q               E  N  S  D  C  S  T  C  C  M  M  A  G  D Q PCT/RAT     P                PARTIAL   SEQUENCE   LIST
STMT SEQ  BF  L  FIELD   SEG-  N  P  E  E  T  U  O  U  N  A  I  V  M  E L  FIELD   RA- C      PICTURE     STRT NO   73 TO 80    SEQ
TYPE NO.  COL F  NAME      LVL  D  R  Q  S  L  B  T  M  T  X  N  G  R  C F  NAME    TIO T       EDIT       CHR CHR
-------------------------------------------------------------------------------------------------------------------------------
 (R1)         (T,TEMP1   )                                                                                        (ASLGEN'D)    37
 (R1)         (T,COUNTER )                                                                                        (ASLGEN'D)    38
 (R1)         (N,EMPNO   ) 001-1                                                                                  (ASLGEN'D)    39
 (R1)         (N,JOB     ) 001-1                                                                                  (ASLGEN'D)    40
 (R1)         (N,LASTNAME) 001-1                                                                                  (ASLGEN'D)    41
 (R1)         (N,FIRSTNME) 001-1                                                                                  (ASLGEN'D)    42
 (R1)         (F,EOF     )                                                                                        (ASLGEN'D)    43
 ** MK4TY05  TYPE 0  SUBFILE RECORD LENGTH COMPUTED FROM THE ABOVE SPECIFICATION IS    86.                        *****
1NOV 05, 2001  12.24.52                                                                                    PAGE      9
                                          ******************************************
                                          *   SUMMARY OF GENERATED SQL STATEMENTS  *
                                          ******************************************

-------------------------------------------------------------------------------------------------------------------------------
FILE>SEGMENT                  SQL STATEMENT                                                  COMMENTS
-------------------------------------------------------------------------------------------------------------------------------
0M4OLD>TEMPL       STATEMENT  1
0          SELECT EMPNO,FIRSTNME,LASTNAME,JOB FROM DSN8610.EMP X1          STANDARD SELECT FOR SEQUENTIAL READ OF ROOT SEGMENT.
                                                                           TABLE ORDERED IN ASCENDING SEQUENCE. DB2 CALL(S): 1
                                                                           PREPARE(S), 1 OPEN(S), AND 49 FETCH(ES).
1NOV 05, 2001  12.24.52                                                                                    PAGE     10
 ** MK4W204  TYPE 0  NUMBER OF MESSAGES PRINTED IS    3.                                                        *****
                     M4OLD    -       49 RECORDS INPUT
                     M4REPO   -       44 RECORDS OUTPUT
                     M4SUBF3  -        4 RECORDS OUTPUT
                     M4INPUT  -       39 RECORDS INPUT
                     M4LIST   -      278 RECORDS OUTPUT
                       15 TRACKS ASSIGNED TO M4LIB    --   12 TRACKS NOT FULL. LIBRARY DIRECTORY BLOCKING FACTOR IS  800.
 ** MK4W209  TYPE 0  1014112 BYTES OF MAIN STORAGE UNUSED DURING DECODING PHASE                                    *****
                     1015000 BYTES OF MAIN STORAGE UNUSED DURING COMPILATION PHASE
                     1060864 BYTES OF MAIN STORAGE UNUSED DURING PROCESSING PHASE
1** MK4ED02  TYPE 0  REQUEST-REPT___3  REPORT-1  REQUESTOR ID-                                                    *****
1                                                                                                        PAGE 1
0-------------------------------------------------------------------------------------------------------------------------------
            TEMP1               COUNTER
-------------------------------------------------------------------------------------------------------------------------------
   ALL DONE                          4
1** MK4FT03  TYPE 0  END OF REPORT.                                                                               *****
1** MK4ED02  TYPE 0  REQUEST-REPORT1   REPORT-1  REQUESTOR ID-                                                    *****
1                                                                                                        PAGE 1
0-------------------------------------------------------------------------------------------------------------------------------
            TEMP1            EMP NO   FIRST NAME      LAST NAME
-------------------------------------------------------------------------------------------------------------------------------
   REPORT ONLY IF JOB IS PRES  000010  CHRISTINE     HAAS
   REPORT ONLY IF JOB IS PRES  123777  STEVE         MILLER
   REPORT ONLY IF JOB IS PRES  123888  ELVIS         PRESLEY
   REPORT ONLY IF JOB IS PRES  666660  BRITNEY       SPEARS
1** MK4FT03  TYPE 0  END OF REPORT.                                                                               *****
1** MK4ED02  TYPE 0  REQUEST-REPORT2   REPORT-1  REQUESTOR ID-                                                    *****
1                                                                                                        PAGE 1
0-------------------------------------------------------------------------------------------------------------------------------
      COUNTER          EOF
-------------------------------------------------------------------------------------------------------------------------------
              1  YNNNNNNNNN
              2  YNNNNNNNNN
              3  YNNNNNNNNN
              4  YNNNNNNNNN
1** MK4FT03  TYPE 0  END OF REPORT.                                                                               *****
1** MK4W204  TYPE 0  NUMBER OF MESSAGES PRINTED IS    6.                                                        *****
                     M4REPO   -       43 RECORDS INPUT
                     M4LIST   -       29 RECORDS OUTPUT
 ** MK4W209  TYPE 0  1059000 BYTES OF MAIN STORAGE UNUSED DURING REPORTING PHASE                                  *****
```

Figure D-1 Sample Application (Page 4 of 4)

# Input File Processing

## Preparation Job

The following item is some of the Output Listing from the Preparation job stream described in Chapter 6, *Input File Processing*. The main reference label and associated member name is R06PREPA.

The Preparation job allocates the files and data sets used to produce the examples shown in Chapter D, *Complete Sample Applications*. The VISION:Builder file definitions are cataloged into a COMLIB library (M4LIB). The files and data sets are loaded with data using VISION:Builder applications or IBM utilities. The actual JCL, source code and data is delivered on the VISION:Builder system tape in the Samples file. This file is downloaded during the installation process. Refer to the *VISION:Builder Installation Guide* for more information.

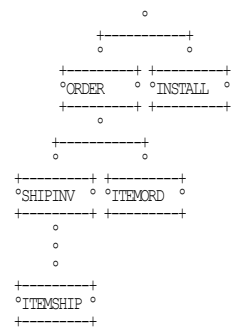
### R06PREPA

```
IDCAMS  SYSTEM SERVICES                                          TIME: 13:19:52       11/03/01     PAGE      1

      DELETE (BUILDER.R140.ITEM.FILE.R06#1) CLUSTER      /* <<< HLQ  */
IDC0550I ENTRY (D) BUILDER.R140.ITEM.FILE.R06#1.DATA DELETED
IDC0550I ENTRY (I) BUILDER.R140.ITEM.FILE.R06#1.INDEX DELETED
IDC0550I ENTRY (C) BUILDER.R140.ITEM.FILE.R06#1 DELETED
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0

      DELETE (BUILDER.R140.CORD.FILE.R06#1) CLUSTER      /* <<< HLQ  */
IDC0550I ENTRY (D) BUILDER.R140.CORD.FILE.R06#1.DATA DELETED
IDC0550I ENTRY (I) BUILDER.R140.CORD.FILE.R06#1.INDEX DELETED
IDC0550I ENTRY (C) BUILDER.R140.CORD.FILE.R06#1 DELETED
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0

      SET MAXCC=0

      DEFINE CLUSTER -
              (NAME('BUILDER.R140.ITEM.FILE.R06#1') -
              INDEXED KEYS (7,0) -
              VOL(ISPDA) -
              TRK(3 1) -
              SHAREOPTIONS(2 3)  -
              RECORDSIZE(38 38)) -
              DATA (CISZ(4096))          /* <<< NAME ... VOL        */
IGD01008I ALLOCATION SET TO IMDCLASS 897
IGD01010I ALLOCATION SET TO IMDVOLS 873
IDC0508I DATA ALLOCATION STATUS FOR VOLUME ISP801 IS 0
IDC0509I INDEX ALLOCATION STATUS FOR VOLUME ISP801 IS 0
IDC0512I NAME GENERATED-(D) BUILDER.R140.ITEM.FILE.R06#1.DATA
IDC0512I NAME GENERATED-(I) BUILDER.R140.ITEM.FILE.R06#1.INDEX
IDC0181I STORAGECLASS USED IS IMDCLASS
IDC0181I MANAGEMENTCLASS USED IS DEFAULT
IDC0181I DATACLASS USED IS DEFAULT
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0
IDCAMS  SYSTEM SERVICES                                          TIME: 13:19:52       11/03/01     PAGE      2

      DEFINE CLUSTER -
              (NAME('BUILDER.R140.CORD.FILE.R06#1') -
              INDEXED KEYS (10,0) -
              VOL(ISPDA) -
              TRK(3 1) -
              SHAREOPTIONS(2 3)  -
              RECORDSIZE(50 50)) -
              DATA (CISZ(4096))         /* <<< NAME ... VOL         */
IGD01008I ALLOCATION SET TO IMDCLASS 897
IGD01010I ALLOCATION SET TO IMDVOLS 873
IDC0508I DATA ALLOCATION STATUS FOR VOLUME ISP803 IS 0
IDC0509I INDEX ALLOCATION STATUS FOR VOLUME ISP803 IS 0
IDC0512I NAME GENERATED-(D) BUILDER.R140.CORD.FILE.R06#1.DATA
IDC0512I NAME GENERATED-(I) BUILDER.R140.CORD.FILE.R06#1.INDEX
IDC0181I STORAGECLASS USED IS IMDCLASS
IDC0181I MANAGEMENTCLASS USED IS DEFAULT
IDC0181I DATACLASS USED IS DEFAULT
```

Figure D-2 R06PREPA (Page 1 of 26)

```
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0

                                    /* <<< A... ... S... IDCAMS2 */
IDC0002I IDCAMS PROCESSING COMPLETE. MAXIMUM CONDITION CODE WAS 0
NOV 03, 2001  13.20.17                                                                                                  PAGE      1
                                          ******************************************
                                          *       COMLIB LIBRARY UTILITIES         *
                                          *       COPYRIGHT (C) 1995, 1997         *
                                          * COMPUTER ASSOCIATES INTERNATIONAL, INC. *
                                          ******************************************

                              DIRECTORY BLOCKING FACTOR:   0   LIBRARY RESERVE: NO   LIBRARY COMPONENT: C4.5
                                        COMPRESSION ACTIVE: NO    COMPRESSION SIZE MINIMUM:   507
** CLSUC01  TYPE 0  THE LIBRARY HAS BEEN INITIALIZED.                                                                    *****
NOV 03, 2001  13.20.17                                                                                                  PAGE      2
** MK4W204  TYPE 0  NUMBER OF MESSAGES PRINTED IS    1.                                                                  *****
                    M4LIST   -       12 RECORDS OUTPUT
                      15 TRACKS ASSIGNED TO M4LIB    --   13 TRACKS NOT FULL. LIBRARY DIRECTORY BLOCKING FACTOR IS  800.
** MK4W209  TYPE 0   825880 BYTES OF MAIN STORAGE UNUSED DURING DICTIONARY MAINTENANCE                                  *****

NOV 03, 2001  13.20.19                                                                                                  PAGE      1
                                          ******************************************
                                          * VISION:Builder   4400  (OS/390 - 14.0)  *
                                          *          COPYRIGHT (C) 2002              *
                                          * COMPUTER ASSOCIATES INTERNATIONAL, INC. *
                                          ******************************************
VISION:Builder JCR1 14.0 110201              ENABLED FOR IBM LANGUAGE ENVIRONMENT                BUILD STAMP = 101306,12:51:24.
                                           COMLIB  MODIFICATIONS (PTFs, APARs) INSTALLED
                                      201,202,203,204,205,206,207,208,209,210,212,213,214,215,216
=============================================================================================================================
=                                                                                                                           =
=                    I N S T A L L A T I O N    P A R A M E T E R S    ( M 4 P A R A M S , M A R K L I B P )               =
=                                                                                                                           =
=  SYSTEM DELIMITER:   #          PAGE HEIGHT:       66         M4LIST WIDTH:      132        DEF WIDTH OF PAGE:  0        =
=                                                                                                                           =
=  AUTO GRAND:         N          HEADING CHAR:      -          SUBTITLE REPEAT:   N          INVALID FIELD:      *        =
=                                                                                                                           =
=  MISSING FIELD:      -          NON-EDIT FIELD:    +          PERCENT CHAR:      %          LEFT SEPARATOR:     (        =
=                                                                                                                           =
=  RIGHT SEPARATOR:    )          SINGLE SEPARATOR:  ,          SOURCE SPACING:    1          PRINT MESSAGES:     Y        =
=                                                                                                                           =
=  CONSOLE MESSAGES:   N          M4REPO BLOCKSIZE:  4,096      INPUT I/O BUFFERS: 2          OUTPUT I/O BUFFERS: 1        =
=                                                                                                                           =
=  SNGL-STEP STORAGE:  8,192      SNGL-STEP SORTSIZE: 524,288   DIGIT SELECT CHAR: 9          ZERO SUPPRESS CHAR: Z        =
=                                                                                                                           =
=  CURRENCY CHAR:      $          PLUS CHAR:         +          MINUS CHAR:        -          CHECK PROTECT CHAR: *        =
=                                                                                                                           =
=  DECIMAL CHAR:       .          GROUPING CHAR:     ,          PRIMARY PLOT CHAR: X          SECONDARY PLOT CHAR:*        =
=                                                                                                                           =
=  FIT PLOT CHAR:      .          HORIZONTAL AXIS:   _          HORIZONTAL HASH:   |          VERTICAL AXIS:      |        =
=                                                                                                                           =
=  VERTICAL HASH:      -          MINUTES/HOUR:      60         SECONDS/MINUTE:    60         TIME DELIMITER:     HH:MM:SS =
=                                                                                                                           =
=  DATE FORMAT:      MM DD, YYYY  TODAY FORMAT+DELIM: MM/DD/YY   ISDATE DELIMITER:  YYYY-MM-DD JULIAN DELIMITER:  YY.DDD    =
=                                                                                                                           =
=  SORT PROGRAM CODE:  2          MINCORE VALUE:     12 K       ALT M4LIST WIDTH:  264        ALT DEF W/OF PAGE:  0        =
=                                                                                                                           =
=  MAX LINES OF TRACE: 1,024      ITEM TRACKING:     0          SUPPRESS NDS REPT?: N         DEFAULT MAXGETMN:   1,024 K  =
=                                                                                                                           =
=  CONDITION CODE 1:   0          CONDITION CODE 2:  4          CONDITION CODE 3:  0          CONDITION CODE 4:   16       =
=                                                                                                                           =
=============================================================================================================================
NOV 03, 2001  13.20.19                                                                                                  PAGE      2
-----------------------------------------------------------------------------------------------------------------------------
   RUN     STMT    FILE    O N S D U T R A B S D S L  RB  R S M R S          START          END           SEQUENCE   LIST
   NAME    TYPE    NAME    L E E I P R E U U R L C S  PL  J O O F S          SEARCH         SEARCH        73 TO 80    SEQ
                           D W Q R D N P D F T M N T  TK  T P P O R
-----------------------------------------------------------------------------------------------------------------------------
(DEFINE  ) (RC)                                                                                                          1
NOV 03, 2001  13.20.19                                                                                                  PAGE      3
** CLSAA00  TYPE 0  DEFINITION CUSTOMER IS BEING PROCESSED.                                                              *****
-----------------------------------------------------------------------------------------------------------------------------
   FILE    STMT    FILE    DELETE GLOSSARY RECORD RECORD RECORDS BUFFER MAPPED     MAPPING        MAPPED        SEQUENCE   LIST
   NAME    TYPE    IDENT    FILE   LEVEL   FORMAT  SIZE  PER BLK  SIZE   FILE    INIT     TERM   RECSIZ        73 TO 80    SEQ
-----------------------------------------------------------------------------------------------------------------------------
(CUSTOMER) (FD)             (R)     (B)     (V)                ( 3000)                                                       2

NOV 03, 2001  13.20.19                                                                                                  PAGE      7
                                                FILE STRUCTURE DIAGRAM FOR                                  DIAGRAM PAGE  1
                                                FILE DEFINITION - CUSTOMER

RECORD FORMAT = VARIABLE BLOCKED                                                  FILE IDENTIFICATION =
RECORD SIZE = 2992    BLOCK SIZE = 3000                                           NUMBER OF SEGMENTS IN FILE = 6

                                                    +---------+
                                                    °CUSTOMER °
                                                    +---------+
```

Figure D-2 R06PREPA (Page 2 of 26)



```
                                                                  °
                                                            +-----------+
                                                            °           °
                                                      +---------+ +---------+
                                                      °ORDER    ° °INSTALL  °
                                                      +---------+ +---------+
                                                            °
                                                      +-----------+
                                                      °           °
                                                +---------+ +---------+
                                                °SHIPINV  ° °ITEMORD  °
                                                +---------+ +---------+
                                                      °
                                                      °
                                                      °
                                                +---------+
                                                °ITEMSHIP °
                                                +---------+
NOV 03, 2001  13.20.19                                                                                    PAGE      8
                                        DETAILED GLOSSARY BY LOCATION FOR                          GLOSSARY PAGE      1
                                           FILE DEFINITION - CUSTOMER
DATE CATALOGED - 11/03/01 13:20:19                                                DATE LAST UPDATED - 11/03/01 13:20:19
CATALOGED BY -                                                                    LAST UPDATED BY -
EXPIRATION DATE -                                                                 DATE LAST USED -
FILE IDENTIFICATION =                                                             RECORD FORMAT = VARIABLE BLOCKED
NUMBER OF SEGMENTS IN FILE = 6                                                    RECORD SIZE = 2992
NUMBER OF FIELDS IN FILE = 45                                                     BLOCK SIZE = 3000

                                         ******************************
                                         *    SEGMENT   1, LEVEL 1    *
                                         *   SEGMENT NAME = CUSTOMER  *
                                         ******************************
SEGMENT OCCURS N TIMES = 1                                        KEY FIELD 1 = CUSTNO     TYPE = C   LENGTH = 5
SEGMENT SIZE = 102
NUMBER OF FIELDS IN SEGMENT = 8
-----------------------------------------------------------------------------------------------------------------------
FIELD   FIELD   FIELD    FIELD   FIELD    DEC    CNT FIELD  EDIT CODES     EDIT    OUTPUT   LINE  DATA BASE NAME/
NAME    TYPE   LOCATION  LENGTH  RNDING  PLACES  FOR SGMT   ( ) ( ) ( )   LENGTH   WIDTH     NO   *** COLUMN HEADING ***
-----------------------------------------------------------------------------------------------------------------------
CUSTNO    C        1       5                                                  5       8
                                                                                                1    ***    CUSTOMER    ***
                                                                                                2    ***     NUMBER     ***
CUSTNAME  C        6      30                                                 30      30
                                                                                                1    ***    CUSTOMER    ***
                                                                                                2    ***      NAME      ***
CUSTPH    C       36      10                                                 10      10
                                                                                                1    ***    CUSTOMER    ***
                                                                                                2    ***     PHONE      ***
CUSTADDR  C       46      25                                                 25      25
                                                                                                1    ***    CUSTOMER    ***
                                                                                                2    ***    ADDRESS     ***
CITY      C       71      23                                                 23      23
STATE     C       94       2                                                  2       5
                                                                                                     ***                ***
                                                                                                2    ***     STATE      ***
CUSTZIP   C       96       5                                                  5       5
                                                                                                1    ***      ZIP       ***
                                                                                                2    ***      CODE      ***
SEG60CNT  F      101       2                        6                         7       7
NOV 03, 2001  13.20.19                                                                                    PAGE      9
                                        DETAILED GLOSSARY BY LOCATION FOR                          GLOSSARY PAGE      2
                                           FILE DEFINITION - CUSTOMER

                                         ******************************
                                         *    SEGMENT   2, LEVEL 2    *
                                         *   SEGMENT NAME = ORDER     *
                                         ******************************
SEGMENT OCCURS N TIMES = 20          SEGMENT ORDER =  ASCENDING     KEY FIELD 1 = ORDERNO    TYPE = C   LENGTH = 5
SEGMENT SIZE = 43
NUMBER OF FIELDS IN SEGMENT = 15
-----------------------------------------------------------------------------------------------------------------------
FIELD   FIELD   FIELD    FIELD   FIELD    DEC    CNT FIELD  EDIT CODES     EDIT    OUTPUT   LINE  DATA BASE NAME/
NAME    TYPE   LOCATION  LENGTH  RNDING  PLACES  FOR SGMT   ( ) ( ) ( )   LENGTH   WIDTH     NO   *** COLUMN HEADING ***
-----------------------------------------------------------------------------------------------------------------------
ORDERNO   C        1       5                                                  5       6
                                                                                                1    ***     ORDER      ***
                                                                                                2    ***     NUMBER     ***
ORDRDATE  C        8       6                                                  6       6
                                                                                                1    ***     ORDER      ***
                                                                                                2    ***     DATE       ***
ORDMONTH  C        8       2                                                  2       5
                                                                                                1    ***     ORDER      ***
                                                                                                2    ***     MONTH      ***
ORDERDAY  C       10       2                                                  2       5
                                                                                                1    ***     ORDER      ***
                                                                                                2    ***      DAY       ***
```

Figure D-2 R06PREPA (Page 3 of 26)

```
ORDRYEAR  C       12       2                                          2       5
                                                                                 1   ***    ORDER     ***
                                                                                 2   ***    YEAR      ***
ORPERSON  C       14       3                                          3       6
                                                                                 1   ***    SALES     ***
                                                                                 2   ***    PERSON    ***
ORDPONUM  C       17       5                                          5      12
                                                                                 1   ***  PURCH. ORDER ***
                                                                                 2   ***    NUMBER    ***
ORDDUDAT  C       22       6                                          6       6
                                                                                 1   ***     DUE      ***
                                                                                 2   ***     DATE     ***
DUEMONTH  C       22       2                                          2       5
                                                                                 1   ***    MONTH     ***
                                                                                 2   ***     DUE      ***
DUEDAY    C       24       2                                          2       3
                                                                                 1   ***     DAY      ***
                                                                                 2   ***     DUE      ***
NOV 03, 2001  13.20.19                                                                                  PAGE     10
                                    DETAILED GLOSSARY BY LOCATION FOR                          GLOSSARY PAGE      3
                                        FILE DEFINITION - CUSTOMER
                                     SEGMENT   2, LEVEL 2 (CONTINUED)
-----------------------------------------------------------------------------------------------------------------------
FIELD   FIELD   FIELD    FIELD   FIELD    DEC    CNT FIELD  EDIT CODES     EDIT    OUTPUT   LINE  DATA BASE NAME/
NAME    TYPE   LOCATION  LENGTH  RNDING  PLACES  FOR SGMT   ( ) ( ) ( )   LENGTH   WIDTH     NO   *** COLUMN HEADING ***
-----------------------------------------------------------------------------------------------------------------------
DUEYEAR   C       26       2                                          2       4
                                                                                 1   ***     YEAR     ***
                                                                                 2   ***     DUE      ***
ORDINVGN  C       28       1                                          1      10
                                                                                 1   ***   INVOICE    ***
                                                                                 2   ***  GENERATED?  ***
ORDCMPLT  C       29       1                                          1      10
                                                                                 1   ***    ORDER     ***
                                                                                 2   ***  COMPLETED?  ***
SEG30CNT  F       40       2                     3                    7       7
SEG50CNT  F       42       2                     5                    7       7
NOV 03, 2001  13.20.19                                                                                  PAGE     11
                                    DETAILED GLOSSARY BY LOCATION FOR                          GLOSSARY PAGE      4
                                        FILE DEFINITION - CUSTOMER

                                     ******************************
                                     *    SEGMENT   3, LEVEL 3    *
                                     *   SEGMENT NAME = SHIPINV   *
                                     ******************************
COUNT FIELD FOR SEGMENT = SEG30CNT     SEGMENT ORDER =  ASCENDING     KEY FIELD 1 = SHIPNO     TYPE = C   LENGTH = 4
SEGMENT SIZE = 100
NUMBER OF FIELDS IN SEGMENT = 11
-----------------------------------------------------------------------------------------------------------------------
FIELD   FIELD   FIELD    FIELD   FIELD    DEC    CNT FIELD  EDIT CODES     EDIT    OUTPUT   LINE  DATA BASE NAME/
NAME    TYPE   LOCATION  LENGTH  RNDING  PLACES  FOR SGMT   ( ) ( ) ( )   LENGTH   WIDTH     NO   *** COLUMN HEADING ***
-----------------------------------------------------------------------------------------------------------------------
SHIPNO    C        1       4                                          4       8
                                                                                 1   ***   SHIPPING    ***
                                                                                 2   ***    NUMBER     ***
SHIPDATE  C        5       6                                          6       7
                                                                                 1   ***    DATE       ***
                                                                                 2   ***   SHIPPED     ***
SHIPPER   C       11       3                                          3       7
                                                                                 1   ***   SHIPPER     ***
FRTCOST   P       23       4            2             ($) ( ) ( )    10      10
                                                                                 1   ***   FREIGHT     ***
                                                                                 2   ***    COST       ***
INVNO     C       41       5                                          5       7
                                                                                 1   ***   INVOICE     ***
                                                                                 2   ***   NUMBER      ***
INVDATE   C       48       6                                          6       7
                                                                                 1   ***   INVOICE     ***
                                                                                 2   ***    DATE       ***
INAMT     P       54       4            2             ($) ( ) ( )    10      10
                                                                                 1   ***   INVOICE     ***
                                                                                 2   ***   AMOUNT      ***
PAYDATE   C       70       6                                          6       7
                                                                                 1   ***   PAYMENT     ***
                                                                                 2   ***    DATE       ***
CHECKNO   C       76       7                                          7       7
                                                                                 1   ***    CHECK      ***
                                                                                 2   ***    NUMBER     ***
PAIDDATE  C       83       6                                          6       7
                                                                                 1   ***   DATE OF     ***
                                                                                 2   ***   PAYMENT     ***
SEG40CNT  F       99       2                     4                    7       7
NOV 03, 2001  13.20.19                                                                                  PAGE     12
                                    DETAILED GLOSSARY BY LOCATION FOR                          GLOSSARY PAGE      5
                                        FILE DEFINITION - CUSTOMER
```

Figure D-2 R06PREPA (Page 4 of 26)

```
                                    ********************************
                                    *    SEGMENT   4, LEVEL 4      *
                                    *   SEGMENT NAME = ITEMSHIP    *
                                    ********************************
COUNT FIELD FOR SEGMENT = SEG40CNT     SEGMENT ORDER =  ASCENDING     KEY FIELD 1 = ITEMSHIP   TYPE = C   LENGTH = 7
SEGMENT SIZE = 14
NUMBER OF FIELDS IN SEGMENT = 2
-------------------------------------------------------------------------------------------------------------------------
FIELD   FIELD   FIELD    FIELD   FIELD    DEC    CNT FIELD  EDIT CODES     EDIT    OUTPUT   LINE  DATA BASE NAME/
NAME    TYPE   LOCATION  LENGTH  RNDING  PLACES  FOR SGMT   ( ) ( ) ( )   LENGTH   WIDTH     NO   *** COLUMN HEADING ***
-------------------------------------------------------------------------------------------------------------------------
ITEMSHIP  C        1       7                                                 7       9
                                                                                      1    ***   STOCK NO.    ***
                                                                                      2    ***    SHIPPED     ***
QTYSHIP   Z        8       7                                                10      10
                                                                                      1    ***    QUANTITY    ***
                                                                                      2    ***    SHIPPED     ***
NOV 03, 2001  13.20.19                                                                                    PAGE     13
                                  DETAILED GLOSSARY BY LOCATION FOR                                GLOSSARY PAGE      6
                                     FILE DEFINITION - CUSTOMER

                                    ********************************
                                    *    SEGMENT   5, LEVEL 3      *
                                    *   SEGMENT NAME = ITEMORD     *
                                    ********************************
COUNT FIELD FOR SEGMENT = SEG50CNT     SEGMENT ORDER =  ASCENDING     KEY FIELD 1 = ITEMORD    TYPE = C   LENGTH = 7
SEGMENT SIZE = 47
NUMBER OF FIELDS IN SEGMENT = 4
-------------------------------------------------------------------------------------------------------------------------
FIELD   FIELD   FIELD    FIELD   FIELD    DEC    CNT FIELD  EDIT CODES     EDIT    OUTPUT   LINE  DATA BASE NAME/
NAME    TYPE   LOCATION  LENGTH  RNDING  PLACES  FOR SGMT   ( ) ( ) ( )   LENGTH   WIDTH     NO   *** COLUMN HEADING ***
-------------------------------------------------------------------------------------------------------------------------
ITEMORD   C        1       7                                                 7       9
                                                                                      1    ***   STOCK NO.    ***
                                                                                      2    ***    ORDERED     ***
ITMPRICE  Z       28       6            2              ($) ( ) ( )          9       9
                                                                                      1    ***     PRICE      ***
ITMQTYOR  Z       34       7                                                10      10
                                                                                      1    ***     ORDER      ***
                                                                                      2    ***    QUANTITY    ***
QTYBKORD  Z       41       7                                                10      11
                                                                                      1    ***    QUANTITY    ***
                                                                                      2    ***  BACKORDERED   ***
NOV 03, 2001  13.20.19                                                                                    PAGE     14
                                  DETAILED GLOSSARY BY LOCATION FOR                                GLOSSARY PAGE      7
                                     FILE DEFINITION - CUSTOMER

                                    ********************************
                                    *    SEGMENT   6, LEVEL 2      *
                                    *   SEGMENT NAME = INSTALL     *
                                    ********************************
COUNT FIELD FOR SEGMENT = SEG60CNT     SEGMENT ORDER =  ASCENDING     KEY FIELD 1 = INSTNO     TYPE = C   LENGTH = 4
SEGMENT SIZE = 25
NUMBER OF FIELDS IN SEGMENT = 5
-------------------------------------------------------------------------------------------------------------------------
FIELD   FIELD   FIELD    FIELD   FIELD    DEC    CNT FIELD  EDIT CODES     EDIT    OUTPUT   LINE  DATA BASE NAME/
NAME    TYPE   LOCATION  LENGTH  RNDING  PLACES  FOR SGMT   ( ) ( ) ( )   LENGTH   WIDTH     NO   *** COLUMN HEADING ***
-------------------------------------------------------------------------------------------------------------------------
INSTNO    C        1       4                                                 4      12
                                                                                      1    ***  INSTALLATION  ***
                                                                                      2    ***    NUMBER      ***
INSTCONT  C        5       3                                                 3      12
                                                                                      1    ***  INSTALLATION  ***
                                                                                      2    ***    CONTACT     ***
INSTOPS   C        8       8                                                 8       9
                                                                                      1    ***   OPERATING    ***
                                                                                      2    ***    SYSTEM      ***
INSTOSRL  C       16       5                                                 5       9
                                                                                      1    ***   OP SYSTEM    ***
                                                                                      2    ***    RELEASE     ***
INPRODID  C       21       5                                                 5       7
                                                                                      1    ***    PRODUCT     ***
                                                                                      2    ***    RELEASE     ***
** CLSGB00  TYPE 0  END OF FILE DEFINITION GLOSSARY.                                                                *****
NOV 03, 2001  13.20.19                                                                                    PAGE     15
** CLSAA00  TYPE 0  DEFINITION CUSTTRNX IS BEING PROCESSED.                                                         *****
-------------------------------------------------------------------------------------------------------------------------
   FILE    STMT    FILE    DELETE GLOSSARY RECORD RECORD RECORDS BUFFER MAPPED     MAPPING         MAPPED          SEQUENCE   LIST
   NAME    TYPE    IDENT    FILE   LEVEL   FORMAT  SIZE  PER BLK  SIZE  FILE    INIT      TERM    RECSIZ         73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------
(CUSTTRNX) (FD)             (R)      (B)      (F)    (  80)                                                            132

NOV 03, 2001  13.20.19                                                                                    PAGE     19
                                  DETAILED GLOSSARY BY LOCATION FOR                                GLOSSARY PAGE      1
                                     FILE DEFINITION - CUSTTRNX
DATE CATALOGED - 11/03/01 13:20:19                                  DATE LAST UPDATED - 11/03/01 13:20:19
```

Figure D-2 R06PREPA (Page 5 of 26)

```
CATALOGED BY -                                                                  LAST UPDATED BY -
EXPIRATION DATE -                                                               DATE LAST USED -
FILE IDENTIFICATION =                                                           RECORD FORMAT = FIXED UNBLOCKED
NUMBER OF SEGMENTS IN FILE = 1                                                  RECORD SIZE = 80
NUMBER OF FIELDS IN FILE = 44                                                   BLOCK SIZE = 80

                                    ********************************
                                    *    SEGMENT   1, LEVEL 1      *
                                    *   SEGMENT NAME = TRANRCD     *
                                    ********************************
SEGMENT OCCURS N TIMES = 1                                  KEY FIELD 1 = CUSTNO     TYPE = C   LENGTH = 5
SEGMENT SIZE = 78
NUMBER OF FIELDS IN SEGMENT = 44
-------------------------------------------------------------------------------------------------------------------------------
FIELD   FIELD   FIELD    FIELD   FIELD    DEC    CNT FIELD  EDIT CODES     EDIT    OUTPUT   LINE  DATA BASE NAME/
NAME    TYPE   LOCATION  LENGTH  RNDING  PLACES  FOR SGMT   ( ) ( ) ( )   LENGTH   WIDTH     NO   *** COLUMN HEADING ***
-------------------------------------------------------------------------------------------------------------------------------
TRANID    C        1       8                                                 8       8
                                                                                          1    ***     TRANS      ***
                                                                                          2    ***      ID        ***
TRANID1   C        1       4                                                 4       5
                                                                                          1    ***     TRANS      ***
                                                                                          2    ***     ID 1       ***
TRANID2   C        5       4                                                 4       5
                                                                                          1    ***     TRANS      ***
                                                                                          2    ***     ID 2       ***
CUSTNO    C        9       5                                                 5       8
                                                                                          1    ***    CUSTOMER    ***
                                                                                          2    ***     NUMBER     ***
CUSTNAME  C       14      30                                                30      30
                                                                                          1    ***    CUSTOMER    ***
                                                                                          2    ***      NAME      ***
INSTNO    C       14       4                                                 4      12
                                                                                          1    ***  INSTALLATION  ***
                                                                                          2    ***     NUMBER     ***
CITY      C       14      23                                                23      23
                                                                                               ***                ***
                                                                                          2    ***      CITY      ***
ORDERNO   C       14       5                                                 5       6
                                                                                          1    ***     ORDER      ***
                                                                                          2    ***     NUMBER     ***
NOV 03, 2001  13.20.19                                                                                          PAGE     20
                                    DETAILED GLOSSARY BY LOCATION FOR                                  GLOSSARY PAGE      2
                                       FILE DEFINITION - CUSTTRNX
                                     SEGMENT   1, LEVEL 1 (CONTINUED)
-------------------------------------------------------------------------------------------------------------------------------
FIELD   FIELD   FIELD    FIELD   FIELD    DEC    CNT FIELD  EDIT CODES     EDIT    OUTPUT   LINE  DATA BASE NAME/
NAME    TYPE   LOCATION  LENGTH  RNDING  PLACES  FOR SGMT   ( ) ( ) ( )   LENGTH   WIDTH     NO   *** COLUMN HEADING ***
-------------------------------------------------------------------------------------------------------------------------------
INSTCONT  C       18       3                                                 3      12
                                                                                          1    ***  INSTALLATION  ***
                                                                                          2    ***    CONTACT     ***
SHIPNO    C       19       4                                                 4       8
                                                                                          1    ***    SHIPPING    ***
                                                                                          2    ***     NUMBER     ***
ORDMONTH  C       19       2                                                 2       5
                                                                                          1    ***     ORDER      ***
                                                                                          2    ***     MONTH      ***
ORDRDATE  C       19       6                                                 6       6
                                                                                          1    ***     ORDER      ***
                                                                                          2    ***     DATE       ***
ITEMORD   C       19       7                                                 7       9
                                                                                          1    ***   STOCK NO.    ***
                                                                                          2    ***    ORDERED     ***
INSTOPS   C       21       8                                                 8       9
                                                                                          1    ***   OPERATING    ***
                                                                                          2    ***    SYSTEM      ***
ORDERDAY  C       21       2                                                 2       5
                                                                                          1    ***     ORDER      ***
                                                                                          2    ***      DAY       ***
ITEMSHIP  C       23       7                                                 7       9
                                                                                          1    ***   STOCK NO.    ***
                                                                                          2    ***    SHIPPED     ***
SHIPDATE  C       23       6                                                 6       7
                                                                                          1    ***     DATE       ***
                                                                                          2    ***    SHIPPED     ***
ORDRYEAR  C       23       2                                                 2       5
                                                                                          1    ***     ORDER      ***
                                                                                          2    ***     YEAR       ***
ORPERSON  C       25       3                                                 3       6
                                                                                          1    ***     SALES      ***
                                                                                          2    ***     PERSON     ***
ORDPONUM  C       28       5                                                 5      12
                                                                                          1    ***  PURCH. ORDER  ***
                                                                                          2    ***     NUMBER     ***
NOV 03, 2001  13.20.19                                                                                          PAGE     21
```

Figure D-2 R06PREPA (Page 6 of 26)

DETAILED GLOSSARY BY LOCATION FOR
FILE DEFINITION - CUSTTRNX
SEGMENT 1, LEVEL 1 (CONTINUED)

GLOSSARY PAGE 3

| FIELD NAME | FIELD TYPE | FIELD LOCATION | FIELD LENGTH | FIELD RNDING | DEC PLACES | CNT FIELD FOR SGMT | EDIT CODES ( ) ( ) ( ) | EDIT LENGTH | OUTPUT WIDTH | LINE NO | DATA BASE NAME/ *** COLUMN HEADING *** |
|---|---|---|---|---|---|---|---|---|---|---|---|
| SHIPPER | C | 29 | 3 | | | | | 3 | 7 | | |
| | | | | | | | | | | | *** *** |
| | | | | | | | | | | 2 | *** SHIPPER *** |
| INSTOSRL | C | 29 | 5 | | | | | 5 | 9 | | |
| | | | | | | | | | | 1 | *** OP SYSTEM *** |
| | | | | | | | | | | 2 | *** RELEASE *** |
| QTYSHIP | Z | 30 | 7 | | | | | 10 | 10 | | |
| | | | | | | | | | | 1 | *** QUANTITY *** |
| | | | | | | | | | | 2 | *** SHIPPED *** |
| FRTCOST | Z | 32 | 7 | | 2 | | ($) ( ) ( ) | 10 | 10 | | |
| | | | | | | | | | | 1 | *** FREIGHT *** |
| | | | | | | | | | | 2 | *** COST *** |
| DUEMONTH | C | 33 | 2 | | | | | 2 | 5 | | |
| | | | | | | | | | | 1 | *** MONTH *** |
| | | | | | | | | | | 2 | *** DUE *** |
| ORDDUDAT | C | 33 | 6 | | | | | 6 | 6 | | |
| | | | | | | | | | | 1 | *** DUE *** |
| | | | | | | | | | | 2 | *** DATE *** |
| INPRODID | C | 34 | 5 | | | | | 5 | 7 | | |
| | | | | | | | | | | 1 | *** PRODUCT *** |
| | | | | | | | | | | 2 | *** RELEASE *** |
| DUEDAY | C | 35 | 2 | | | | | 2 | 3 | | |
| | | | | | | | | | | 1 | *** DAY *** |
| | | | | | | | | | | 2 | *** DUE *** |
| DUEYEAR | C | 37 | 2 | | | | | 2 | 4 | | |
| | | | | | | | | | | 1 | *** YEAR *** |
| | | | | | | | | | | 2 | *** DUE *** |
| STATE | C | 37 | 2 | | | | | 2 | 5 | | |
| | | | | | | | | | | | *** *** |
| | | | | | | | | | | 2 | *** STATE *** |
| ORDINVGN | C | 39 | 1 | | | | | 1 | 10 | | |
| | | | | | | | | | | 1 | *** INVOICE *** |
| | | | | | | | | | | 2 | *** GENERATED? *** |
| CUSTZIP | C | 39 | 5 | | | | | 5 | 5 | | |
| | | | | | | | | | | 1 | *** ZIP *** |
| | | | | | | | | | | 2 | *** CODE *** |



DETAILED GLOSSARY BY LOCATION FOR
FILE DEFINITION - CUSTTRNX
SEGMENT 1, LEVEL 1 (CONTINUED)

GLOSSARY PAGE 4

| FIELD NAME | FIELD TYPE | FIELD LOCATION | FIELD LENGTH | FIELD RNDING | DEC PLACES | CNT FIELD FOR SGMT | EDIT CODES ( ) ( ) ( ) | EDIT LENGTH | OUTPUT WIDTH | LINE NO | DATA BASE NAME/ *** COLUMN HEADING *** |
|---|---|---|---|---|---|---|---|---|---|---|---|
| INVNO | C | 39 | 5 | | | | | 5 | 7 | | |
| | | | | | | | | | | 1 | *** INVOICE *** |
| | | | | | | | | | | 2 | *** NUMBER *** |
| ORDCMPLT | C | 40 | 1 | | | | | 1 | 10 | | |
| | | | | | | | | | | 1 | *** ORDER *** |
| | | | | | | | | | | 2 | *** COMPLETED? *** |
| CUSTPH | C | 44 | 10 | | | | | 10 | 10 | | |
| | | | | | | | | | | 1 | *** CUSTOMER *** |
| | | | | | | | | | | 2 | *** PHONE *** |
| INVDATE | C | 44 | 6 | | | | | 6 | 7 | | |
| | | | | | | | | | | 1 | *** INVOICE *** |
| | | | | | | | | | | 2 | *** DATE *** |
| ITMPRICE | Z | 46 | 6 | | 2 | | ($) ( ) ( ) | 9 | 9 | | |
| | | | | | | | | | | 1 | *** PRICE *** |
| INAMT | Z | 50 | 7 | | 2 | | ($) ( ) ( ) | 10 | 10 | | |
| | | | | | | | | | | 1 | *** INVOICE *** |
| | | | | | | | | | | 2 | *** AMOUNT *** |
| ITMQTYOR | Z | 52 | 7 | | | | | 10 | 10 | | |
| | | | | | | | | | | 1 | *** ORDER *** |
| | | | | | | | | | | 2 | *** QUANTITY *** |
| CUSTADDR | C | 54 | 25 | | | | | 25 | 25 | | |
| | | | | | | | | | | 1 | *** CUSTOMER *** |
| | | | | | | | | | | 2 | *** ADDRESS *** |
| PAYDATE | C | 57 | 6 | | | | | 6 | 7 | | |
| | | | | | | | | | | 1 | *** PAYMENT *** |
| | | | | | | | | | | 2 | *** DATE *** |
| QTYBKORD | Z | 59 | 7 | | | | | 10 | 11 | | |
| | | | | | | | | | | 1 | *** QUANTITY *** |
| | | | | | | | | | | 2 | *** BACKORDERED *** |
| CHECKNO | C | 63 | 7 | | | | | 7 | 7 | | |
| | | | | | | | | | | 1 | *** CHECK *** |
| | | | | | | | | | | 2 | *** NUMBER *** |
| PAIDDATE | C | 70 | 6 | | | | | 6 | 7 | | |
| | | | | | | | | | | 1 | *** DATE OF *** |
| | | | | | | | | | | 2 | *** PAYMENT *** |

Figure D-2 R06PREPA (Page 7 of 26)

```
** CLSGB00  TYPE 0  END OF FILE DEFINITION GLOSSARY.                                                                          *****
NOV 03, 2001  13.20.19                                                                                                     PAGE    23
** CLSAA00  TYPE 0  DEFINITION CUSTTRAN IS BEING PROCESSED.                                                                   *****
-----------------------------------------------------------------------------------------------------------------------------------
  GROUP     STMT IDENT1 IDENT1 ID1 IDENT2 IDENT2 ID2 DEL GLS    FILE    DEFLT DEFLT REC RECORD RECORDS BUFFER          SEQUENCE  LIST
  NAME      TYPE  CODE  LOCATN LNG  CODE  LOCATN LNG ETE ARY    NAME    CREAT INSRT FRM  SIZE  PER BLK  SIZE           73 TO 80   SEQ
-----------------------------------------------------------------------------------------------------------------------------------
(CUSTTRAN) (TD) (1ADD) (   1) (4) (CUST) (   5) (4) (R)     (CUSTOMER)             (F) (  80)                                    274
COMMENT ***** TRANSACTION DEFINITION FOR "CUSTOMER" FILE                      *****                                              275
COMMENT *****                                                                 *****                                              276
COMMENT ***** ADD CUSTOMER TRANSACTION 1                                      *****                                              277
(CUSTTRAN) (TD) (2ADD) (   1) (4) (CUST) (   5) (4) (R)     (CUSTOMER)             (F) (  80)                                    282
COMMENT ***** ADD CUSTOMER TRANSACTION 2                                      *****                                              283
(CUSTTRAN) (TD) (3ADD) (   1) (4) (ORDR) (   5) (4) (R)     (CUSTOMER)             (F) (  80)                                    288
COMMENT ***** ADD ORDER TRANSACTION                                           *****                                              289
(CUSTTRAN) (TD) (4ADD) (   1) (4) (ITEM) (   5) (4) (R)     (CUSTOMER)             (F) (  80)                                    298
COMMENT ***** ADD ITEM TO ORDER TRANSACTION                                   *****                                              299
(CUSTTRAN) (TD) (5ADD) (   1) (4) (INVO) (   5) (4) (R)     (CUSTOMER)             (F) (  80)                                    306
COMMENT ***** ADD SHIPPING INVOICE TRANSACTION                                *****                                              307
(CUSTTRAN) (TD) (6ADD) (   1) (4) (SHIP) (   5) (4) (R)     (CUSTOMER)             (F) (  80)                                    320
COMMENT ***** ADD SHIPPING ITEM TRANSACTION                                   *****                                              321
(CUSTTRAN) (TD) (7ADD) (   1) (4) (INST) (   5) (4) (R)     (CUSTOMER)             (F) (  80)                                    327
COMMENT ***** ADD INSTALLATION TRANSACTION                                    *****                                              328
(CUSTTRAN) (TD) (1UPD) (   1) (4) (CUST) (   5) (4) (R)     (CUSTOMER)             (F) (  80)                                    335
COMMENT ***** UPD CUSTOMER TRANSACTION 1                                      *****                                              336
(CUSTTRAN) (TD) (2UPD) (   1) (4) (CUST) (   5) (4) (R)     (CUSTOMER)             (F) (  80)                                    341
COMMENT ***** UPD CUSTOMER TRANSACTION 2                                      *****                                              342
(CUSTTRAN) (TD) (3UPD) (   1) (4) (ORDR) (   5) (4) (R)     (CUSTOMER)             (F) (  80)                                    347
COMMENT ***** UPD ORDER TRANSACTION                                           *****                                              348
(CUSTTRAN) (TD) (4UPD) (   1) (4) (ITEM) (   5) (4) (R)     (CUSTOMER)             (F) (  80)                                    357
COMMENT ***** UPD ITEM TO ORDER TRANSACTION                                   *****                                              358
(CUSTTRAN) (TD) (5UPD) (   1) (4) (INVO) (   5) (4) (R)     (CUSTOMER)             (F) (  80)                                    365
COMMENT ***** UPD SHIPPING INVOICE TRANSACTION                                *****                                              366
(CUSTTRAN) (TD) (6UPD) (   1) (4) (SHIP) (   5) (4) (R)     (CUSTOMER)             (F) (  80)                                    379
COMMENT ***** UPD SHIPPING ITEM TRANSACTION                                   *****                                              380
(CUSTTRAN) (TD) (7UPD) (   1) (4) (INST) (   5) (4) (R)     (CUSTOMER)             (F) (  80)                                    386
COMMENT ***** UPD INSTALLATION TRANSACTION                                    *****                                              387
(CUSTTRAN) (TD) (1DEL) (   1) (4) (ORDR) (   5) (4) (R)     (CUSTOMER)             (F) (  80)                                    394
COMMENT ***** DELETE ORDER TRANSACTION                                        *****                                              395
(CUSTTRAN) (TD) (2DEL) (   1) (4) (ITEM) (   5) (4) (R)     (CUSTOMER)             (F) (  80)                                    398
COMMENT ***** DELETE ITEM FROM ORDER TRANSACTION                              *****                                              399
(CUSTTRAN) (TD) (3DEL) (   1) (4) (CUST) (   5) (4) (R)     (CUSTOMER)             (F) (  80)                                    403
COMMENT ***** DELETE CUSTOMER TRANSACTION                                     *****                                              404

NOV 03, 2001  13.20.19                                                                                                     PAGE    28
                                             GLOSSARY BY LEVEL NUMBER FOR                                          GLOSSARY PAGE    1
                                             TRANSACTION GROUP - CUSTTRAN

DATE CATALOGED - 11/03/01 13:20:19                                                 DATE LAST UPDATED - 11/03/01 13:20:19
CATALOGED BY -                                                                     LAST UPDATED BY -
EXPIRATION DATE -                                                                  DATE LAST USED -
MASTER FILE NAME = CUSTOMER                                                        RECORD FORMAT = FIXED UNBLOCKED
NUMBER OF IDENTIFIERS IN GROUP = 17                                                RECORD SIZE = 80
NUMBER OF ACTIONS IN GROUP = 96                                                    BLOCK SIZE = 80

                                        **************************************
                                        *  1  OF  17 IDENTIFIERS IN GROUP   *
                                        **************************************
**********    ACTIONS IN IDENTIFIER = 4             IDENTIFIER ONE = (1ADD)             IDENTIFIER TWO = (CUST)
*1ADDCUST*    DEFAULT CREATE = NO                   IDENTIFIER LOCATION = 1             IDENTIFIER LOCATION = 5
**********    DEFAULT INSERT = NO                   IDENTIFIER LENGTH = 4               IDENTIFIER LENGTH = 4
                                                    IDENTIFIED IF EQUAL                 IDENTIFIED IF EQUAL
-----------------------------------------------------------------------------------------------------------------------
MASTER  ..........TRANSACTION DATA..........
FILE    FIELD     FIELD     FIELD    DECIMAL   ACTION   LEVEL          MINIMUM                   MAXIMUM
FIELD   TYPE    LOCATION   LENGTH    PLACES     CODE    NUMBER          VALUE                     VALUE
-----------------------------------------------------------------------------------------------------------------------
CUSTNO    C          9        5                  C         1        EDIT = Y9999
CUSTNAME  C         14       30                  R         1
CUSTPH    C         44       10                  R         1
CUSTADDR  C         54       25                  R         1
NOV 03, 2001  13.20.19                                                                                                     PAGE    29
                                             GLOSSARY BY LEVEL NUMBER FOR                                          GLOSSARY PAGE    2
                                             TRANSACTION GROUP - CUSTTRAN

                                        **************************************
                                        *  2  OF  17 IDENTIFIERS IN GROUP   *
                                        **************************************
**********    ACTIONS IN IDENTIFIER = 4             IDENTIFIER ONE = (2ADD)             IDENTIFIER TWO = (CUST)
*2ADDCUST*    DEFAULT CREATE = NO                   IDENTIFIER LOCATION = 1             IDENTIFIER LOCATION = 5
**********    DEFAULT INSERT = NO                   IDENTIFIER LENGTH = 4               IDENTIFIER LENGTH = 4
                                                    IDENTIFIED IF EQUAL                 IDENTIFIED IF EQUAL
-----------------------------------------------------------------------------------------------------------------------
MASTER  ..........TRANSACTION DATA..........
FILE    FIELD     FIELD     FIELD    DECIMAL   ACTION   LEVEL          MINIMUM                   MAXIMUM
FIELD   TYPE    LOCATION   LENGTH    PLACES     CODE    NUMBER          VALUE                     VALUE
-----------------------------------------------------------------------------------------------------------------------
```

Figure D-2    R06PREPA  (Page 8 of 26)

```
CUSTNO    C          9       5              M        1          EDIT = Y9999
CITY      C         14      23              R        1
STATE     C         37       2              R        1
CUSTZIP   C         39       5              R        1
NOV 03, 2001  13.20.19                                                                                  PAGE     30
                                    GLOSSARY BY LEVEL NUMBER FOR                                GLOSSARY PAGE      3
                                    TRANSACTION GROUP - CUSTTRAN

                                 ***************************************
                                 *   3  OF  17 IDENTIFIERS IN GROUP   *
                                 ***************************************
**********    ACTIONS IN IDENTIFIER = 8          IDENTIFIER ONE = (3ADD)          IDENTIFIER TWO = (ORDR)
*3ADDORDR*    DEFAULT CREATE = NO                IDENTIFIER LOCATION = 1          IDENTIFIER LOCATION = 5
**********    DEFAULT INSERT = NO                IDENTIFIER LENGTH = 4            IDENTIFIER LENGTH = 4
                                                 IDENTIFIED IF EQUAL              IDENTIFIED IF EQUAL
-----------------------------------------------------------------------------------------------------------------
MASTER  ..........TRANSACTION DATA..........
FILE    FIELD     FIELD     FIELD    DECIMAL   ACTION   LEVEL          MINIMUM                MAXIMUM
FIELD   TYPE     LOCATION   LENGTH   PLACES     CODE    NUMBER          VALUE                  VALUE
-----------------------------------------------------------------------------------------------------------------
CUSTNO    C          9       5              M        1          EDIT = Y9999
ORDERNO   C         14       5              I        2
ORDRDATE  C         19       6              R        2
ORPERSON  C         25       3              R        2          EDIT = -9-9-9
ORDPONUM  C         28       5              R        2
ORDDUDAT  C         33       6              R        2
ORDINVGN  C         39       1              R        2          EDIT = Z
ORDCMPLT  C         40       1              R        2
NOV 03, 2001  13.20.19                                                                                  PAGE     31
                                    GLOSSARY BY LEVEL NUMBER FOR                                GLOSSARY PAGE      4
                                    TRANSACTION GROUP - CUSTTRAN

                                 ***************************************
                                 *   4  OF  17 IDENTIFIERS IN GROUP   *
                                 ***************************************
**********    ACTIONS IN IDENTIFIER = 6          IDENTIFIER ONE = (4ADD)          IDENTIFIER TWO = (ITEM)
*4ADDITEM*    DEFAULT CREATE = NO                IDENTIFIER LOCATION = 1          IDENTIFIER LOCATION = 5
**********    DEFAULT INSERT = NO                IDENTIFIER LENGTH = 4            IDENTIFIER LENGTH = 4
                                                 IDENTIFIED IF EQUAL              IDENTIFIED IF EQUAL
-----------------------------------------------------------------------------------------------------------------
MASTER  ..........TRANSACTION DATA..........
FILE    FIELD     FIELD     FIELD    DECIMAL   ACTION   LEVEL          MINIMUM                MAXIMUM
FIELD   TYPE     LOCATION   LENGTH   PLACES     CODE    NUMBER          VALUE                  VALUE
-----------------------------------------------------------------------------------------------------------------
CUSTNO    C          9       5              M        1          EDIT = Y9999
ORDERNO   C         14       5              M        2
ITEMORD   C         19       7              I        3
ITMPRICE  Z         46       6       2      R        3     ***0.00          ***     ***999.99        ***
ITMQTYOR  Z         52       7              R        3
QTYBKORD  Z         59       7              R        3
NOV 03, 2001  13.20.19                                                                                  PAGE     32
                                    GLOSSARY BY LEVEL NUMBER FOR                                GLOSSARY PAGE      5
                                    TRANSACTION GROUP - CUSTTRAN

                                 ***************************************
                                 *   5  OF  17 IDENTIFIERS IN GROUP   *
                                 ***************************************
**********    ACTIONS IN IDENTIFIER = 12         IDENTIFIER ONE = (5ADD)          IDENTIFIER TWO = (INVO)
*5ADDINVO*    DEFAULT CREATE = NO                IDENTIFIER LOCATION = 1          IDENTIFIER LOCATION = 5
**********    DEFAULT INSERT = NO                IDENTIFIER LENGTH = 4            IDENTIFIER LENGTH = 4
                                                 IDENTIFIED IF EQUAL              IDENTIFIED IF EQUAL
-----------------------------------------------------------------------------------------------------------------
MASTER  ..........TRANSACTION DATA..........
FILE    FIELD     FIELD     FIELD    DECIMAL   ACTION   LEVEL          MINIMUM                MAXIMUM
FIELD   TYPE     LOCATION   LENGTH   PLACES     CODE    NUMBER          VALUE                  VALUE
-----------------------------------------------------------------------------------------------------------------
CUSTNO    C          9       5              M        1          EDIT = Y9999
ORDERNO   C         14       5              M        2
SHIPNO    C         19       4              I        3
SHIPDATE  C         23       6              R        3
SHIPPER   C         29       3              R        3
FRTCOST   Z         32       7       2      R        3
INVNO     C         39       5              R        3
INVDATE   C         44       6              R        3
INAMT     Z         50       7       2      R        3
PAYDATE   C         57       6              R        3
CHECKNO   C         63       7              R        3
PAIDDATE  C         70       6              R        3
NOV 03, 2001  13.20.19                                                                                  PAGE     33
                                    GLOSSARY BY LEVEL NUMBER FOR                                GLOSSARY PAGE      6
                                    TRANSACTION GROUP - CUSTTRAN

                                 ***************************************
                                 *   6  OF  17 IDENTIFIERS IN GROUP   *
                                 ***************************************
**********    ACTIONS IN IDENTIFIER = 5          IDENTIFIER ONE = (6ADD)          IDENTIFIER TWO = (SHIP)
```

Figure D-2 R06PREPA (Page 9 of 26)

```
*6ADDSHIP*    DEFAULT CREATE = NO                IDENTIFIER LOCATION = 1          IDENTIFIER LOCATION = 5
**********    DEFAULT INSERT = NO                IDENTIFIER LENGTH = 4            IDENTIFIER LENGTH = 4
                                                 IDENTIFIED IF EQUAL              IDENTIFIED IF EQUAL
-------------------------------------------------------------------------------------------------------------------
MASTER  .........TRANSACTION DATA.........
FILE    FIELD     FIELD     FIELD    DECIMAL   ACTION   LEVEL          MINIMUM                MAXIMUM
FIELD   TYPE    LOCATION   LENGTH   PLACES     CODE    NUMBER          VALUE                  VALUE
-------------------------------------------------------------------------------------------------------------------
CUSTNO    C          9        5                M        1          EDIT = Y9999
ORDERNO   C         14        5                M        2
SHIPNO    C         19        4                M        3
ITEMSHIP  C         23        7                I        4
QTYSHIP   Z         30        7                R        4
NOV 03, 2001  13.20.19                                                                          PAGE     34
                                        GLOSSARY BY LEVEL NUMBER FOR                    GLOSSARY PAGE     7
                                        TRANSACTION GROUP - CUSTTRAN

                                     ************************************
                                     *  7  OF  17 IDENTIFIERS IN GROUP   *
                                     ************************************
**********    ACTIONS IN IDENTIFIER = 6          IDENTIFIER ONE = (7ADD)          IDENTIFIER TWO = (INST)
*7ADDINST*    DEFAULT CREATE = NO                IDENTIFIER LOCATION = 1          IDENTIFIER LOCATION = 5
**********    DEFAULT INSERT = NO                IDENTIFIER LENGTH = 4            IDENTIFIER LENGTH = 4
                                                 IDENTIFIED IF EQUAL              IDENTIFIED IF EQUAL
-------------------------------------------------------------------------------------------------------------------
MASTER  .........TRANSACTION DATA.........
FILE    FIELD     FIELD     FIELD    DECIMAL   ACTION   LEVEL          MINIMUM                MAXIMUM
FIELD   TYPE    LOCATION   LENGTH   PLACES     CODE    NUMBER          VALUE                  VALUE
-------------------------------------------------------------------------------------------------------------------
CUSTNO    C          9        5                M        1          EDIT = Y9999
INSTNO    C         14        4                I        2
INSTCONT  C         18        3                R        2
INSTOPS   C         21        8                R        2
INSTOSRL  C         29        5                R        2
INPRODID  C         34        5                R        2
NOV 03, 2001  13.20.19                                                                          PAGE     35
                                        GLOSSARY BY LEVEL NUMBER FOR                    GLOSSARY PAGE     8
                                        TRANSACTION GROUP - CUSTTRAN

                                     ************************************
                                     *  8  OF  17 IDENTIFIERS IN GROUP   *
                                     ************************************
**********    ACTIONS IN IDENTIFIER = 4          IDENTIFIER ONE = (1UPD)          IDENTIFIER TWO = (CUST)
*1UPDCUST*    DEFAULT CREATE = NO                IDENTIFIER LOCATION = 1          IDENTIFIER LOCATION = 5
**********    DEFAULT INSERT = NO                IDENTIFIER LENGTH = 4            IDENTIFIER LENGTH = 4
                                                 IDENTIFIED IF EQUAL              IDENTIFIED IF EQUAL
-------------------------------------------------------------------------------------------------------------------
MASTER  .........TRANSACTION DATA.........
FILE    FIELD     FIELD     FIELD    DECIMAL   ACTION   LEVEL          MINIMUM                MAXIMUM
FIELD   TYPE    LOCATION   LENGTH   PLACES     CODE    NUMBER          VALUE                  VALUE
-------------------------------------------------------------------------------------------------------------------
CUSTNO    C          9        5                M        1          EDIT = Y9999
CUSTNAME  C         14       30                R        1
CUSTPH    C         44       10                R        1
CUSTADDR  C         54       25                R        1
NOV 03, 2001  13.20.19                                                                          PAGE     36
                                        GLOSSARY BY LEVEL NUMBER FOR                    GLOSSARY PAGE     9
                                        TRANSACTION GROUP - CUSTTRAN

                                     ************************************
                                     *  9  OF  17 IDENTIFIERS IN GROUP   *
                                     ************************************
**********    ACTIONS IN IDENTIFIER = 4          IDENTIFIER ONE = (2UPD)          IDENTIFIER TWO = (CUST)
*2UPDCUST*    DEFAULT CREATE = NO                IDENTIFIER LOCATION = 1          IDENTIFIER LOCATION = 5
**********    DEFAULT INSERT = NO                IDENTIFIER LENGTH = 4            IDENTIFIER LENGTH = 4
                                                 IDENTIFIED IF EQUAL              IDENTIFIED IF EQUAL
-------------------------------------------------------------------------------------------------------------------
MASTER  .........TRANSACTION DATA.........
FILE    FIELD     FIELD     FIELD    DECIMAL   ACTION   LEVEL          MINIMUM                MAXIMUM
FIELD   TYPE    LOCATION   LENGTH   PLACES     CODE    NUMBER          VALUE                  VALUE
-------------------------------------------------------------------------------------------------------------------
CUSTNO    C          9        5                M        1          EDIT = Y9999
CITY      C         14       23                R        1
STATE     C         37        2                R        1
CUSTZIP   C         39        5                R        1
NOV 03, 2001  13.20.19                                                                          PAGE     37
                                        GLOSSARY BY LEVEL NUMBER FOR                    GLOSSARY PAGE    10
                                        TRANSACTION GROUP - CUSTTRAN

                                     ************************************
                                     *  10 OF  17 IDENTIFIERS IN GROUP   *
                                     ************************************
**********    ACTIONS IN IDENTIFIER = 8          IDENTIFIER ONE = (3UPD)          IDENTIFIER TWO = (ORDR)
*3UPDORDR*    DEFAULT CREATE = NO                IDENTIFIER LOCATION = 1          IDENTIFIER LOCATION = 5
**********    DEFAULT INSERT = NO                IDENTIFIER LENGTH = 4            IDENTIFIER LENGTH = 4
                                                 IDENTIFIED IF EQUAL              IDENTIFIED IF EQUAL
```

Figure D-2 R06PREPA (Page 10 of 26)

```
-----------------------------------------------------------------------------------------------------------------
MASTER  ..........TRANSACTION DATA..........
FILE    FIELD     FIELD     FIELD    DECIMAL   ACTION   LEVEL          MINIMUM                  MAXIMUM
FIELD   TYPE     LOCATION   LENGTH   PLACES     CODE    NUMBER          VALUE                    VALUE
-----------------------------------------------------------------------------------------------------------------
CUSTNO    C          9        5                  M        1          EDIT = Y9999
ORDERNO   C         14        5                  M        2
ORDRDATE  C         19        6                  R        2
ORPERSON  C         25        3                  R        2          EDIT = -9-9-9
ORDPONUM  C         28        5                  R        2
ORDDUDAT  C         33        6                  R        2
ORDINVGN  C         39        1                  R        2          EDIT = Z
ORDCMPLT  C         40        1                  R        2
NOV 03, 2001  13.20.19                                                                                  PAGE     38
                                          GLOSSARY BY LEVEL NUMBER FOR                                GLOSSARY PAGE     11
                                          TRANSACTION GROUP - CUSTTRAN

                                     ***************************************
                                     *   11 OF  17 IDENTIFIERS IN GROUP   *
                                     ***************************************
**********    ACTIONS IN IDENTIFIER = 6          IDENTIFIER ONE = (4UPD)          IDENTIFIER TWO = (ITEM)
*4UPDITEM*    DEFAULT CREATE = NO                IDENTIFIER LOCATION = 1          IDENTIFIER LOCATION = 5
**********    DEFAULT INSERT = NO                IDENTIFIER LENGTH = 4            IDENTIFIER LENGTH = 4
                                                 IDENTIFIED IF EQUAL              IDENTIFIED IF EQUAL
-----------------------------------------------------------------------------------------------------------------
MASTER  ..........TRANSACTION DATA..........
FILE    FIELD     FIELD     FIELD    DECIMAL   ACTION   LEVEL          MINIMUM                  MAXIMUM
FIELD   TYPE     LOCATION   LENGTH   PLACES     CODE    NUMBER          VALUE                    VALUE
-----------------------------------------------------------------------------------------------------------------
CUSTNO    C          9        5                  M        1          EDIT = Y9999
ORDERNO   C         14        5                  M        2
ITEMORD   C         19        7                  I        3
ITMPRICE  Z         47        6        2         R        3     ***0.00          ***     ***300.0         ***
ITMQTYOR  Z         60        7                  R        3
QTYBKORD  Z         67        7                  R        3
NOV 03, 2001  13.20.19                                                                                  PAGE     39
                                          GLOSSARY BY LEVEL NUMBER FOR                                GLOSSARY PAGE     12
                                          TRANSACTION GROUP - CUSTTRAN

                                     ***************************************
                                     *   12 OF  17 IDENTIFIERS IN GROUP   *
                                     ***************************************
**********    ACTIONS IN IDENTIFIER = 12         IDENTIFIER ONE = (5UPD)          IDENTIFIER TWO = (INVO)
*5UPDINVO*    DEFAULT CREATE = NO                IDENTIFIER LOCATION = 1          IDENTIFIER LOCATION = 5
**********    DEFAULT INSERT = NO                IDENTIFIER LENGTH = 4            IDENTIFIER LENGTH = 4
                                                 IDENTIFIED IF EQUAL              IDENTIFIED IF EQUAL
-----------------------------------------------------------------------------------------------------------------
MASTER  ..........TRANSACTION DATA..........
FILE    FIELD     FIELD     FIELD    DECIMAL   ACTION   LEVEL          MINIMUM                  MAXIMUM
FIELD   TYPE     LOCATION   LENGTH   PLACES     CODE    NUMBER          VALUE                    VALUE
-----------------------------------------------------------------------------------------------------------------
CUSTNO    C          9        5                  M        1          EDIT = Y9999
ORDERNO   C         14        5                  M        2
SHIPNO    C         19        4                  I        3
SHIPDATE  C         23        6                  R        3
SHIPPER   C         29        3                  R        3
FRTCOST   Z         32        7        2         R        3
INVNO     C         39        5                  R        3
INVDATE   C         44        6                  R        3
INAMT     Z         50        7        2         R        3
PAYDATE   C         57        6                  R        3
CHECKNO   C         63        7                  R        3
PAIDDATE  C         70        6                  R        3
NOV 03, 2001  13.20.19                                                                                  PAGE     40
                                          GLOSSARY BY LEVEL NUMBER FOR                                GLOSSARY PAGE     13
                                          TRANSACTION GROUP - CUSTTRAN

                                     ***************************************
                                     *   13 OF  17 IDENTIFIERS IN GROUP   *
                                     ***************************************
**********    ACTIONS IN IDENTIFIER = 5          IDENTIFIER ONE = (6UPD)          IDENTIFIER TWO = (SHIP)
*6UPDSHIP*    DEFAULT CREATE = NO                IDENTIFIER LOCATION = 1          IDENTIFIER LOCATION = 5
**********    DEFAULT INSERT = NO                IDENTIFIER LENGTH = 4            IDENTIFIER LENGTH = 4
                                                 IDENTIFIED IF EQUAL              IDENTIFIED IF EQUAL
-----------------------------------------------------------------------------------------------------------------
MASTER  ..........TRANSACTION DATA..........
FILE    FIELD     FIELD     FIELD    DECIMAL   ACTION   LEVEL          MINIMUM                  MAXIMUM
FIELD   TYPE     LOCATION   LENGTH   PLACES     CODE    NUMBER          VALUE                    VALUE
-----------------------------------------------------------------------------------------------------------------
CUSTNO    C          9        5                  M        1          EDIT = Y9999
ORDERNO   C         14        5                  M        2
SHIPNO    C         19        4                  M        3
ITEMSHIP  C         23        7                  M        4
QTYSHIP   Z         30        7                  R        4
NOV 03, 2001  13.20.19                                                                                  PAGE     41
                                          GLOSSARY BY LEVEL NUMBER FOR                                GLOSSARY PAGE     14
```

Figure D-2 R06PREPA (Page 11 of 26)

```
                                    TRANSACTION GROUP - CUSTTRAN

                                  ************************************
                                  *  14 OF  17 IDENTIFIERS IN GROUP   *
                                  ************************************
**********    ACTIONS IN IDENTIFIER = 6          IDENTIFIER ONE = (7UPD)          IDENTIFIER TWO = (INST)
*7UPDINST*    DEFAULT CREATE = NO                IDENTIFIER LOCATION = 1          IDENTIFIER LOCATION = 5
**********    DEFAULT INSERT = NO                IDENTIFIER LENGTH = 4            IDENTIFIER LENGTH = 4
                                                 IDENTIFIED IF EQUAL              IDENTIFIED IF EQUAL
-------------------------------------------------------------------------------------------------------------------------------
MASTER  .........TRANSACTION DATA.........
FILE    FIELD     FIELD     FIELD    DECIMAL   ACTION   LEVEL          MINIMUM                MAXIMUM
FIELD   TYPE     LOCATION   LENGTH   PLACES     CODE    NUMBER          VALUE                  VALUE
-------------------------------------------------------------------------------------------------------------------------------
CUSTNO    C          9        5                 M        1          EDIT = Y9999
INSTNO    C         14        4                 M        2
INSTCONT  C         18        3                 R        2
INSTOPS   C         21        8                 R        2
INSTOSRL  C         29        5                 R        2
INPRODID  C         34        5                 R        2
NOV 03, 2001  13.20.19                                                                                    PAGE     42
                                    GLOSSARY BY LEVEL NUMBER FOR                                  GLOSSARY PAGE     15
                                    TRANSACTION GROUP - CUSTTRAN

                                  ************************************
                                  *  15 OF  17 IDENTIFIERS IN GROUP   *
                                  ************************************
**********    ACTIONS IN IDENTIFIER = 2          IDENTIFIER ONE = (1DEL)          IDENTIFIER TWO = (ORDR)
*1DELORDR*    DEFAULT CREATE = NO                IDENTIFIER LOCATION = 1          IDENTIFIER LOCATION = 5
**********    DEFAULT INSERT = NO                IDENTIFIER LENGTH = 4            IDENTIFIER LENGTH = 4
                                                 IDENTIFIED IF EQUAL              IDENTIFIED IF EQUAL
-------------------------------------------------------------------------------------------------------------------------------
MASTER  .........TRANSACTION DATA.........
FILE    FIELD     FIELD     FIELD    DECIMAL   ACTION   LEVEL          MINIMUM                MAXIMUM
FIELD   TYPE     LOCATION   LENGTH   PLACES     CODE    NUMBER          VALUE                  VALUE
-------------------------------------------------------------------------------------------------------------------------------
CUSTNO    C          9        5                 M        1          EDIT = Y9999
ORDERNO   C         14        5                 E        2
NOV 03, 2001  13.20.19                                                                                    PAGE     43
                                    GLOSSARY BY LEVEL NUMBER FOR                                  GLOSSARY PAGE     16
                                    TRANSACTION GROUP - CUSTTRAN

                                  ************************************
                                  *  16 OF  17 IDENTIFIERS IN GROUP   *
                                  ************************************
**********    ACTIONS IN IDENTIFIER = 3          IDENTIFIER ONE = (2DEL)          IDENTIFIER TWO = (ITEM)
*2DELITEM*    DEFAULT CREATE = NO                IDENTIFIER LOCATION = 1          IDENTIFIER LOCATION = 5
**********    DEFAULT INSERT = NO                IDENTIFIER LENGTH = 4            IDENTIFIER LENGTH = 4
                                                 IDENTIFIED IF EQUAL              IDENTIFIED IF EQUAL
-------------------------------------------------------------------------------------------------------------------------------
MASTER  .........TRANSACTION DATA.........
FILE    FIELD     FIELD     FIELD    DECIMAL   ACTION   LEVEL          MINIMUM                MAXIMUM
FIELD   TYPE     LOCATION   LENGTH   PLACES     CODE    NUMBER          VALUE                  VALUE
-------------------------------------------------------------------------------------------------------------------------------
CUSTNO    C          9        5                 M        1          EDIT = Y9999
ORDERNO   C         14        5                 M        2
ITEMORD   C         19        7                 E        3
NOV 03, 2001  13.20.19                                                                                    PAGE     44
                                    GLOSSARY BY LEVEL NUMBER FOR                                  GLOSSARY PAGE     17
                                    TRANSACTION GROUP - CUSTTRAN

                                  ************************************
                                  *  17 OF  17 IDENTIFIERS IN GROUP   *
                                  ************************************
**********    ACTIONS IN IDENTIFIER = 1          IDENTIFIER ONE = (3DEL)          IDENTIFIER TWO = (CUST)
*3DELCUST*    DEFAULT CREATE = NO                IDENTIFIER LOCATION = 1          IDENTIFIER LOCATION = 5
**********    DEFAULT INSERT = NO                IDENTIFIER LENGTH = 4            IDENTIFIER LENGTH = 4
                                                 IDENTIFIED IF EQUAL              IDENTIFIED IF EQUAL
-------------------------------------------------------------------------------------------------------------------------------
MASTER  .........TRANSACTION DATA.........
FILE    FIELD     FIELD     FIELD    DECIMAL   ACTION   LEVEL          MINIMUM                MAXIMUM
FIELD   TYPE     LOCATION   LENGTH   PLACES     CODE    NUMBER          VALUE                  VALUE
-------------------------------------------------------------------------------------------------------------------------------
CUSTNO    C          9        5                 D        1          EDIT = Y9999
** CLSGT00  TYPE 0  END OF TRANSACTION DEFINITION GLOSSARY.                                                                *****

NOV 03, 2001  13.20.19                                                                                    PAGE     45
** CLSAA00  TYPE 0  DEFINITION CUSTHIST IS BEING PROCESSED.                                                               *****
-------------------------------------------------------------------------------------------------------------------------------
   FILE    STMT    FILE    DELETE GLOSSARY RECORD RECORD RECORDS BUFFER MAPPED      MAPPING         MAPPED          SEQUENCE   LIST
   NAME    TYPE    IDENT    FILE   LEVEL   FORMAT  SIZE  PER BLK  SIZE   FILE    INIT      TERM    RECSIZ          73 TO 80   SEQ
-------------------------------------------------------------------------------------------------------------------------------
(CUSTHIST) (FD)             (R)      (B)      (F)    (  80)                                                                406

NOV 03, 2001  13.20.19                                                                                    PAGE     46
                                    DETAILED GLOSSARY BY LOCATION FOR                             GLOSSARY PAGE      1
```

Figure D-2 R06PREPA (Page 12 of 26)

```
                                    FILE DEFINITION - CUSTHIST
DATE CATALOGED - 11/03/01 13:20:19                                  DATE LAST UPDATED - 11/03/01 13:20:19
CATALOGED BY -                                                      LAST UPDATED BY -
EXPIRATION DATE -                                                   DATE LAST USED -
FILE IDENTIFICATION =                                               RECORD FORMAT = FIXED UNBLOCKED
NUMBER OF SEGMENTS IN FILE = 1                                      RECORD SIZE = 80
NUMBER OF FIELDS IN FILE = 5                                        BLOCK SIZE = 80

                                    ********************************
                                    *    SEGMENT   1, LEVEL 1     *
                                    *   SEGMENT NAME = CUSTOMER   *
                                    ********************************
SEGMENT OCCURS N TIMES = 1                                  KEY FIELD 1 = CUSTNO     TYPE = C   LENGTH = 5
SEGMENT SIZE = 24                                           KEY FIELD 2 = ORDERNO    TYPE = C   LENGTH = 5
NUMBER OF FIELDS IN SEGMENT = 5
-------------------------------------------------------------------------------------------------------------------------------
FIELD   FIELD   FIELD    FIELD   FIELD    DEC    CNT FIELD  EDIT CODES     EDIT    OUTPUT   LINE  DATA BASE NAME/
NAME    TYPE   LOCATION LENGTH  RNDING  PLACES  FOR SGMT   ( ) ( ) ( )   LENGTH   WIDTH     NO  *** COLUMN HEADING ***
-------------------------------------------------------------------------------------------------------------------------------
CUSTNO    C        1       5                                               5        8
                                                                                      1   ***    CUSTOMER    ***
                                                                                      2   ***    NUMBER      ***
ORDERNO   C        6       5                                               5        6
                                                                                      1   ***    ORDER       ***
                                                                                      2   ***    NUMBER      ***
ORDRDATE  C       11       6                                               6        6
                                                                                      1   ***    ORDER       ***
                                                                                      2   ***    DATE        ***
ORPERSON  C       17       3                                               3        6
                                                                                      1   ***    SALES       ***
                                                                                      2   ***    PERSON      ***
ORDPONUM  C       20       5                                               5       12
                                                                                      1   ***  PURCH. ORDER  ***
                                                                                      2   ***    NUMBER      ***
** CLSGB00  TYPE 0  END OF FILE DEFINITION GLOSSARY.                                                            *****
NOV 03, 2001  13.20.19                                                                                PAGE     47
** CLSAA00  TYPE 0  DEFINITION ITEMHIST IS BEING PROCESSED.                                                     *****
-------------------------------------------------------------------------------------------------------------------------------
   FILE    STMT    FILE    DELETE GLOSSARY RECORD RECORD RECORDS BUFFER MAPPED      MAPPING         MAPPED          SEQUENCE   LIST
   NAME    TYPE    IDENT    FILE   LEVEL   FORMAT  SIZE  PER BLK  SIZE   FILE    INIT      TERM     RECSIZ          73 TO 80   SEQ
-------------------------------------------------------------------------------------------------------------------------------
(ITEMHIST) (FD)             (R)      (B)      (F)    (  80)                                                          423

NOV 03, 2001  13.20.19                                                                                PAGE     48
                                    DETAILED GLOSSARY BY LOCATION FOR                                 GLOSSARY PAGE      1
                                       FILE DEFINITION - ITEMHIST
DATE CATALOGED - 11/03/01 13:20:19                                  DATE LAST UPDATED - 11/03/01 13:20:19
CATALOGED BY -                                                      LAST UPDATED BY -
EXPIRATION DATE -                                                   DATE LAST USED -
FILE IDENTIFICATION =                                               RECORD FORMAT = FIXED UNBLOCKED
NUMBER OF SEGMENTS IN FILE = 1                                      RECORD SIZE = 80
NUMBER OF FIELDS IN FILE = 5                                        BLOCK SIZE = 80

                                    ********************************
                                    *    SEGMENT   1, LEVEL 1     *
                                    *   SEGMENT NAME = CUSTOMER   *
                                    ********************************
SEGMENT OCCURS N TIMES = 1                                  KEY FIELD 1 = CUSTNO     TYPE = C   LENGTH = 5
SEGMENT SIZE = 34                                           KEY FIELD 2 = ORDERNO    TYPE = C   LENGTH = 5
NUMBER OF FIELDS IN SEGMENT = 5                             KEY FIELD 3 = ITEMNO     TYPE = C   LENGTH = 7
-------------------------------------------------------------------------------------------------------------------------------
FIELD   FIELD   FIELD    FIELD   FIELD    DEC    CNT FIELD  EDIT CODES     EDIT    OUTPUT   LINE  DATA BASE NAME/
NAME    TYPE   LOCATION LENGTH  RNDING  PLACES  FOR SGMT   ( ) ( ) ( )   LENGTH   WIDTH     NO  *** COLUMN HEADING ***
-------------------------------------------------------------------------------------------------------------------------------
CUSTNO    C        1       5                                               5        8
                                                                                      1   ***    CUSTOMER    ***
                                                                                      2   ***    NUMBER      ***
ORDERNO   C        7       5                                               5        6
                                                                                      1   ***    ORDER       ***
                                                                                      2   ***    NUMBER      ***
ITEMNO    C       13       7                                               7        7
                                                                                      1   ***     ITEM       ***
                                                                                      2   ***    NUMBER      ***
ITMPRICE  Z       21       6            2              ($) ( ) ( )        9        9
                                                                                      1   ***    ITEM        ***
                                                                                      2   ***    PRICE       ***
ITEMQTY   Z       28       7                                              10       10
                                                                                      1   ***    ORDER       ***
                                                                                      2   ***   QUANTITY     ***
** CLSGB00  TYPE 0  END OF FILE DEFINITION GLOSSARY.                                                            *****
NOV 03, 2001  13.20.19                                                                                PAGE     49
** CLSAA00  TYPE 0  DEFINITION ITEMFILE IS BEING PROCESSED.                                                     *****
-------------------------------------------------------------------------------------------------------------------------------
   FILE    STMT    FILE    DELETE GLOSSARY RECORD RECORD RECORDS BUFFER MAPPED      MAPPING         MAPPED          SEQUENCE   LIST
   NAME    TYPE    IDENT    FILE   LEVEL   FORMAT  SIZE  PER BLK  SIZE   FILE    INIT      TERM     RECSIZ          73 TO 80   SEQ
-------------------------------------------------------------------------------------------------------------------------------
```

Figure D-2 R06PREPA (Page 13 of 26)

```
(ITEMFILE) (FD)             (R)             (K)                 (   38)                                                          440
COMMENT ***** ITEM FILE CONTAINS ITEM NAME AND WEIGHT                        *****                                                 441

NOV 03, 2001  13.20.19                                                                                                  PAGE     50
                                      DETAILED GLOSSARY BY NAME FOR                                                GLOSSARY PAGE      1
                                        FILE DEFINITION - ITEMFILE
DATE CATALOGED - 11/03/01 13:20:19                                    DATE LAST UPDATED - 11/03/01 13:20:19
CATALOGED BY -                                                        LAST UPDATED BY -
EXPIRATION DATE -                                                     DATE LAST USED -
FILE IDENTIFICATION =                                                 RECORD FORMAT = KEY SEQUENCED VSAM
NUMBER OF SEGMENTS IN FILE = 1                                        RECORD SIZE = 38
NUMBER OF FIELDS IN FILE = 4                                          BUFFER SIZE = 38

                                     *********************************
                                     *    SEGMENT   1, LEVEL 1     *
                                     *   SEGMENT NAME = ITEMSEG    *
                                     *********************************
SEGMENT OCCURS N TIMES = 1                                  KEY FIELD 1 = ITEMNO     TYPE = C   LENGTH = 7
SEGMENT SIZE = 38
NUMBER OF FIELDS IN SEGMENT = 4
SEGMENT KEY CHARACTERISTICS = U
-------------------------------------------------------------------------------------------------------------------------------
FIELD   FIELD   FIELD    FIELD   FIELD    DEC    CNT FIELD  EDIT CODES     EDIT    OUTPUT   LINE  DATA BASE NAME/
NAME    TYPE   LOCATION  LENGTH  RNDING  PLACES  FOR SGMT   ( ) ( ) ( )   LENGTH  WIDTH     NO   *** COLUMN HEADING ***
-------------------------------------------------------------------------------------------------------------------------------
ITEMNAME  C        8       20                                               20      20
                                                                                      1   ***      ITEM      ***
                                                                                      2   ***      NAME      ***
ITEMNO    C        1       7                                                 7       7
                                                                                      1   ***      ITEM      ***
                                                                                      2   ***     NUMBER     ***
ITMPRICE  Z       28       6              2              ($) ( ) ( )       9       9
                                                                                      1   ***      ITEM      ***
                                                                                      2   ***     PRICE      ***
ITMWEIGH  Z       34       5              2                                 7       7
                                                                                      1   ***      ITEM      ***
                                                                                      2   ***     WEIGH      ***
** CLSGB00  TYPE 0  END OF FILE DEFINITION GLOSSARY.                                                                          *****
NOV 03, 2001  13.20.19                                                                                                  PAGE     51
** CLSAA00  TYPE 0  DEFINITION ITEMDAT1 IS BEING PROCESSED.                                                                  *****
-------------------------------------------------------------------------------------------------------------------------------
   FILE    STMT    FILE    DELETE GLOSSARY RECORD RECORD RECORDS BUFFER MAPPED     MAPPING        MAPPED          SEQUENCE   LIST
   NAME    TYPE    IDENT    FILE   LEVEL   FORMAT  SIZE  PER BLK  SIZE  FILE    INIT     TERM    RECSIZ         73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(ITEMDAT1) (FD)             (R)             (F)    (   80)                                                                   455
COMMENT ***** ITEM DAT1 CONTAINS DATA TO CREATE THE ITEM FILE                *****                                            456

NOV 03, 2001  13.20.19                                                                                                  PAGE     52
                                      DETAILED GLOSSARY BY NAME FOR                                                GLOSSARY PAGE      1
                                        FILE DEFINITION - ITEMDAT1
DATE CATALOGED - 11/03/01 13:20:19                                    DATE LAST UPDATED - 11/03/01 13:20:19
CATALOGED BY -                                                        LAST UPDATED BY -
EXPIRATION DATE -                                                     DATE LAST USED -
FILE IDENTIFICATION =                                                 RECORD FORMAT = FIXED UNBLOCKED
NUMBER OF SEGMENTS IN FILE = 1                                        RECORD SIZE = 80
NUMBER OF FIELDS IN FILE = 4                                          BLOCK SIZE = 80

                                     *********************************
                                     *    SEGMENT   1, LEVEL 1     *
                                     *   SEGMENT NAME = ITEMSEG    *
                                     *********************************
SEGMENT OCCURS N TIMES = 1                                  KEY FIELD 1 = ITEMNO     TYPE = C   LENGTH = 7
SEGMENT SIZE = 41
NUMBER OF FIELDS IN SEGMENT = 4
-------------------------------------------------------------------------------------------------------------------------------
FIELD   FIELD   FIELD    FIELD   FIELD    DEC    CNT FIELD  EDIT CODES     EDIT    OUTPUT   LINE  DATA BASE NAME/
NAME    TYPE   LOCATION  LENGTH  RNDING  PLACES  FOR SGMT   ( ) ( ) ( )   LENGTH  WIDTH     NO   *** COLUMN HEADING ***
-------------------------------------------------------------------------------------------------------------------------------
ITEMNAME  C        9       20                                               20      20
                                                                                      1   ***      ITEM      ***
                                                                                      2   ***      NAME      ***
ITEMNO    C        1       7                                                 7       7
                                                                                      1   ***      ITEM      ***
                                                                                      2   ***     NUMBER     ***
ITMPRICE  Z       30       6              2              ($) ( ) ( )       9       9
                                                                                      1   ***      ITEM      ***
                                                                                      2   ***     PRICE      ***
ITMWEIGH  Z       37       5              2                                 7       7
                                                                                      1   ***      ITEM      ***
                                                                                      2   ***     WEIGH      ***
** CLSGB00  TYPE 0  END OF FILE DEFINITION GLOSSARY.                                                                          *****
NOV 03, 2001  13.20.19                                                                                                  PAGE     53
** CLSAA00  TYPE 0  DEFINITION MASTER1  IS BEING PROCESSED.                                                                  *****
-------------------------------------------------------------------------------------------------------------------------------
   FILE    STMT    FILE    DELETE GLOSSARY RECORD RECORD RECORDS BUFFER MAPPED     MAPPING        MAPPED          SEQUENCE   LIST
   NAME    TYPE    IDENT    FILE   LEVEL   FORMAT  SIZE  PER BLK  SIZE  FILE    INIT     TERM    RECSIZ         73 TO 80    SEQ
```

Figure D-2 R06PREPA (Page 14 of 26)

```
--------------------------------------------------------------------------------------------------------------------------------
(MASTER1 ) (FD)             (R)      (B)      (F)    (0050)                                                              470

NOV 03, 2001  13.20.19                                                                                          PAGE     54
                                    DETAILED GLOSSARY BY LOCATION FOR                                  GLOSSARY PAGE      1
                                       FILE DEFINITION - MASTER1
DATE CATALOGED - 11/03/01 13:20:19                                  DATE LAST UPDATED - 11/03/01 13:20:19
CATALOGED BY -                                                      LAST UPDATED BY -
EXPIRATION DATE -                                                   DATE LAST USED -
FILE IDENTIFICATION =                                               RECORD FORMAT = FIXED UNBLOCKED
NUMBER OF SEGMENTS IN FILE = 1                                      RECORD SIZE = 50
NUMBER OF FIELDS IN FILE = 2                                        BLOCK SIZE = 50

                                     *******************************
                                     *    SEGMENT   1, LEVEL 1     *
                                     *******************************
SEGMENT OCCURS N TIMES = 1                                KEY FIELD 1 = CUSTNUM    TYPE = C   LENGTH = 5
SEGMENT SIZE = 50
NUMBER OF FIELDS IN SEGMENT = 2
--------------------------------------------------------------------------------------------------------------------------------
FIELD   FIELD   FIELD    FIELD   FIELD    DEC    CNT FIELD  EDIT CODES     EDIT    OUTPUT   LINE  DATA BASE NAME/
NAME    TYPE   LOCATION  LENGTH  RNDING  PLACES  FOR SGMT   ( ) ( ) ( )   LENGTH   WIDTH     NO  *** COLUMN HEADING ***
--------------------------------------------------------------------------------------------------------------------------------
CUSTNUM   C        1       5                                                  5       8
                                                                                      1   ***    CUSTOMER    ***
                                                                                      2   ***     NUMBER     ***
DATA      C        6       45                                                45       45
                                                                                          ***                ***
                                                                                      2   ***   MSTR DATA    ***
** CLSGB00  TYPE 0  END OF FILE DEFINITION GLOSSARY.                                                              *****
NOV 03, 2001  13.20.19                                                                                          PAGE     55
** CLSAA00  TYPE 0  DEFINITION CORDFILE IS BEING PROCESSED.                                                       *****
--------------------------------------------------------------------------------------------------------------------------------
   FILE    STMT    FILE    DELETE GLOSSARY RECORD RECORD RECORDS BUFFER MAPPED      MAPPING         MAPPED          SEQUENCE   LIST
   NAME    TYPE    IDENT    FILE   LEVEL   FORMAT  SIZE  PER BLK  SIZE   FILE    INIT      TERM     RECSIZ          73 TO 80   SEQ
--------------------------------------------------------------------------------------------------------------------------------
(CORDFILE) (FD)             (R)      (B)      (K)                 (    50)                                                476

NOV 03, 2001  13.20.19                                                                                          PAGE     56
                                    DETAILED GLOSSARY BY LOCATION FOR                                  GLOSSARY PAGE      1
                                       FILE DEFINITION - CORDFILE
DATE CATALOGED - 11/03/01 13:20:19                                  DATE LAST UPDATED - 11/03/01 13:20:19
CATALOGED BY -                                                      LAST UPDATED BY -
EXPIRATION DATE -                                                   DATE LAST USED -
FILE IDENTIFICATION =                                               RECORD FORMAT = KEY SEQUENCED VSAM
NUMBER OF SEGMENTS IN FILE = 1                                      RECORD SIZE = 50
NUMBER OF FIELDS IN FILE = 5                                        BUFFER SIZE = 50

                                     *******************************
                                     *    SEGMENT   1, LEVEL 1     *
                                     *   SEGMENT NAME = ORDRSEG    *
                                     *******************************
SEGMENT OCCURS N TIMES = 1                                KEY FIELD 1 = CUSTORDR   TYPE = C   LENGTH = 10
SEGMENT SIZE = 50
NUMBER OF FIELDS IN SEGMENT = 5
SEGMENT KEY CHARACTERISTICS = U
--------------------------------------------------------------------------------------------------------------------------------
FIELD   FIELD   FIELD    FIELD   FIELD    DEC    CNT FIELD  EDIT CODES     EDIT    OUTPUT   LINE  DATA BASE NAME/
NAME    TYPE   LOCATION  LENGTH  RNDING  PLACES  FOR SGMT   ( ) ( ) ( )   LENGTH   WIDTH     NO  *** COLUMN HEADING ***
--------------------------------------------------------------------------------------------------------------------------------
CUSTORDR  C        1       10                                                 10      12
                                                                                      1   ***  CUSTOMER &    ***
                                                                                      2   ***  ORDER NUMBER  ***
CUSTNUM   C        1       5                                                  5       6
                                                                                      1   ***     CUSTNO     ***
                                                                                      2   ***      ONLY      ***
CUST04    C        1       4                                                  4       7
                                                                                      1   ***     CUSTNO     ***
                                                                                      2   ***     POS 1-4    ***
ORDERNUM  C        6       5                                                  5       7
                                                                                      1   ***     ORDERNO    ***
                                                                                      2   ***      ONLY      ***
DATA      C       11       40                                                40       40
                                                                                          ***                ***
                                                                                      2   ***    CORD DATA   ***
** CLSGB00  TYPE 0  END OF FILE DEFINITION GLOSSARY.                                                              *****
NOV 03, 2001  13.20.19                                                                                          PAGE     57
** MK4W204  TYPE 0  NUMBER OF MESSAGES PRINTED IS   85.                                                          *****
                    M4INPUT  -      491 RECORDS INPUT
                    M4LIST   -     1936 RECORDS OUTPUT
                      15 TRACKS ASSIGNED TO M4LIB    --   13 TRACKS NOT FULL. LIBRARY DIRECTORY BLOCKING FACTOR IS  800.
** MK4W209  TYPE 0   924904 BYTES OF MAIN STORAGE UNUSED DURING DICTIONARY MAINTENANCE                            *****

NOV 03, 2001  13.20.21                                                                                          PAGE      1
                                     *****************************************
                                     * VISION:Builder   4400  (OS/390 - 14.0)  *
```

Figure D-2 R06PREPA (Page 15 of 26)

```
                                        *           COPYRIGHT (C) 2002           *
                                        *  COMPUTER ASSOCIATES INTERNATIONAL, INC. *
                                        ********************************************
VISION:Builder JCR1 14.0 110201                ENABLED FOR IBM LANGUAGE ENVIRONMENT                BUILD STAMP = 101306,12:51:24.
                                             COMLIB  MODIFICATIONS (PTFs, APARs) INSTALLED
                                       201,202,203,204,205,206,207,208,209,210,212,213,214,215,216
=============================================================================================================================
=                                                                                                                           =
=                    I N S T A L L A T I O N    P A R A M E T E R S    ( M 4 P A R A M S , M A R K L I B P )               =
=                                                                                                                           =
=  SYSTEM DELIMITER:   #          PAGE HEIGHT:       66         M4LIST WIDTH:      132        DEF WIDTH OF PAGE:  0        =
=                                                                                                                           =
=  AUTO GRAND:         N          HEADING CHAR:      -          SUBTITLE REPEAT:   N          INVALID FIELD:      *        =
=                                                                                                                           =
=  MISSING FIELD:      -          NON-EDIT FIELD:    +          PERCENT CHAR:      %          LEFT SEPARATOR:     (        =
=                                                                                                                           =
=  RIGHT SEPARATOR:    )          SINGLE SEPARATOR:  ,          SOURCE SPACING:    1          PRINT MESSAGES:     Y        =
=                                                                                                                           =
=  CONSOLE MESSAGES:   N          M4REPO BLOCKSIZE:  4,096      INPUT I/O BUFFERS: 2          OUTPUT I/O BUFFERS: 1        =
=                                                                                                                           =
=  SNGL-STEP STORAGE:  8,192      SNGL-STEP SORTSIZE: 524,288   DIGIT SELECT CHAR: 9          ZERO SUPPRESS CHAR: Z        =
=                                                                                                                           =
=  CURRENCY CHAR:      $          PLUS CHAR:         +          MINUS CHAR:        -          CHECK PROTECT CHAR: *        =
=                                                                                                                           =
=  DECIMAL CHAR:       .          GROUPING CHAR:     ,          PRIMARY PLOT CHAR: X          SECONDARY PLOT CHAR:*        =
=                                                                                                                           =
=  FIT PLOT CHAR:      .          HORIZONTAL AXIS:   _          HORIZONTAL HASH:   |          VERTICAL AXIS:      |        =
=                                                                                                                           =
=  VERTICAL HASH:      -          MINUTES/HOUR:      60         SECONDS/MINUTE:    60         TIME DELIMITER:     HH:MM:SS =
=                                                                                                                           =
=  DATE FORMAT:      MM DD, YYYY  TODAY FORMAT+DELIM: MM/DD/YY   ISDATE DELIMITER:  YYYY-MM-DD JULIAN DELIMITER:  YY.DDD    =
=                                                                                                                           =
=  SORT PROGRAM CODE:  2          MINCORE VALUE:     12 K       ALT M4LIST WIDTH:  264        ALT DEF W/OF PAGE:  0        =
=                                                                                                                           =
=  MAX LINES OF TRACE: 1,024      ITEM TRACKING:     0          SUPPRESS NDS REPT?: N         DEFAULT MAXGEIMN:   1,024 K  =
=                                                                                                                           =
=  CONDITION CODE 1:   0          CONDITION CODE 2:  4          CONDITION CODE 3:  0          CONDITION CODE 4:   16       =
=                                                                                                                           =
=============================================================================================================================
NOV 03, 2001  13.20.21                                                                                              PAGE     2
-----------------------------------------------------------------------------------------------------------------------------
   RUN     STMT    FILE    O N S D U T R A B S D S L  RB  R S M R S          START          END            SEQUENCE   LIST
   NAME    TYPE    NAME    L E E I P R E U U R L C S  PL  J O O F S          SEARCH         SEARCH         73 TO 80    SEQ
                           D W Q R D N P D F T M N T  TK  T P P O R
-----------------------------------------------------------------------------------------------------------------------------
(BLDCUSTM) (RC) (CUSTOMER)  (S)     (S,S)       (T,G)         (R)                                        (ASLGEN'D)     1
    CONTROL NAME BLDCUSTM TERM LISTGEN
    FILE MASTER OUTPUT CUSTOMER
    FILE TRAN   GROUPS CUSTTRAN
    FILE REPORT
-----------------------------------------------------------------------------------------------------------------------------
   RUN     STMT  LOGICAL  LBL  DTE/DD     KEY       KEY       KEY     QUALIFIER   USER I/O  ARRAY      PASSWORD/  SEQUENCE   LIST
   NAME    TYPE  FILENAME       NAME     FIELD-1   FIELD-2   FIELD-3  ICF CRD DBI   MODULE   QUAL       AUTH.-ID   73 TO 80    SEQ
-----------------------------------------------------------------------------------------------------------------------------
(BLDCUSTM) (RF) (CUSTOMER) (S) (M4NEW   )                                                                (ASLGEN'D)     2
(BLDCUSTM) (RF)            (S) (M4TRAN  )                                                                (ASLGEN'D)     3
-----------------------------------------------------------------------------------------------------------------------------
   RUN     STMT                TRANSACTION GROUPS                                                       SEQUENCE   LIST
   NAME    TYPE     1        2        3        4        5                                               73 TO 80    SEQ
-----------------------------------------------------------------------------------------------------------------------------
(BLDCUSTM) (RG) (CUSTTRAN)                                                                               (ASLGEN'D)     4


=============================================================================================================================
=                                                                                                                           =
=                                          F I L E    S U M M A R Y                                                         =
=                                                                                                                           =
= JOBNAME: KREJO04X                        STEPNAME: PROC1                                                                  =
=                                                                                                                           =
=  FILE NAME: M4NEW                        DSNAME: BUILDER.R140.CUSTOMER.MASTER.R06#1                                       =
=  FILE NAME: M4TRAN                       DSNAME: KREJO04.KREJO04X.JOB04981.D0000103.?                                     =
=  FILE NAME: M4LIB                        DSNAME: BUILDER.R140.M4LIB.R06#1                                                 =
=                                                                                                                           =
=============================================================================================================================
NOV 03, 2001  13.20.21                                                                                              PAGE     3
                                            ******************************
                                            *  PROC NAME - REPT   1      *
                                            *  INPUT STREAM PROCEDURE    *
                                            ******************************
-----------------------------------------------------------------------------------------------------------------------------
STMT REPORT                          MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID    ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM       73 TO 80    SEQ
-----------------------------------------------------------------------------------------------------------------------------
(ER) (TODAY )                                                                (N)                                (ASLGEN'D)     5
     REPORT CUSTNO CUSTNAME CUSTPH CUSTADDR CITY STATE CUSTZIP SEG60CNT
```

Figure D-2 R06PREPA (Page 16 of 26)

```
      FORMAT HEADINGS NAME
      TITLE 'BUILD THE CUSTOMER MASTER FILE'
      TITLE 'LIST THE CUSTOMER SEGMENT DATA - FIELD NAMES AS HEADINGS'
    END REPORT
--------------------------------------------------------------------------------------------------------------------------------
 ST SUM V  8  PG  PG   I SP  MAXIMUM TB COLUMN  DT  PG  STRT LIN SL RH .......SUBFILE........ C E A S R S N      C  SEQUENCE   LIST
 TY RPT S LPI WID HGT  M FRM LPP PGS TO HEADING POS POS PAGE NUM UB PN ENT    NAME   BLK/ FMT H F G T S U D      O  73 TO 80    SEQ
 MP     P             G           LT TYP POS        NUM      ML TD REC          BUF      C C S L T M S      N
--------------------------------------------------------------------------------------------------------------------------------
(E1)                                (F)                                                                      (ASLGEN'D)     6
--------------------------------------------------------------------------------------------------------------------------------
        SP  Q              E N S D C S T C C M M A G D Q PCT/RAT     P              PARTIAL   SEQUENCE   LIST
STMT SEQ BF  L  FIELD   SEG- N P E E T U O U N A I V M E L FIELD   RA- C     PICTURE     STRT NO  73 TO 80    SEQ
TYPE NO. COL F  NAME     LVL D R Q S L B T M T X N G R C F NAME    TIO T      EDIT       CHR CHR
--------------------------------------------------------------------------------------------------------------------------------
(R1)          (CUSTNO  ) 001-1                                                                               (ASLGEN'D)     7
(R1)          (CUSTNAME) 001-1                                                                               (ASLGEN'D)     8
(R1)          (CUSTPH  ) 001-1                                                                               (ASLGEN'D)     9
(R1)          (CUSTADDR) 001-1                                                                               (ASLGEN'D)    10
(R1)          (CITY    ) 001-1                                                                               (ASLGEN'D)    11
(R1)          (STATE   ) 001-1                                                                               (ASLGEN'D)    12
(R1)          (CUSTZIP ) 001-1                                                                               (ASLGEN'D)    13
(R1)          (SEG60CNT) 001-1                                                                               (ASLGEN'D)    14
--------------------------------------------------------------------------------------------------------------------------------
STMT  SEQ                                                                                                     SEQUENCE   LIST
TYPE  NO.                                        TEXT                                                         73 TO 80    SEQ
--------------------------------------------------------------------------------------------------------------------------------
(T1) (101)              (BUILD THE CUSTOMER MASTER FILE#                          )                           (ASLGEN'D)    15
(T1) (102)              (LIST THE CUSTOMER SEGMENT DATA - FIELD NAMES AS HEADINGS# )                           (ASLGEN'D)    16
NOV 03, 2001  13.20.21                                                                                          PAGE     4
                                                  ******************************
                                                  *  PROC NAME - REPT   2      *
                                                  *  INPUT STREAM PROCEDURE    *
                                                  ******************************
--------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                          MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID    ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM       73 TO 80    SEQ
--------------------------------------------------------------------------------------------------------------------------------
(ER) (TODAY )                                                          (N)                                    (ASLGEN'D)    17
    REPORT CUSTNO,
           ORDERNO ORDRDATE ORPERSON ORDPONUM ORDDUDAT ORDINVGN,
           ORDCMPLT SEG30CNT SEG50CNT
      FORMAT HEADINGS NAME
      TITLE 'BUILD THE CUSTOMER MASTER FILE'
      TITLE 'LIST THE ORDER SEGMENT DATA - FIELD NAMES AS HEADINGS'
    END REPORT
--------------------------------------------------------------------------------------------------------------------------------
 ST SUM V  8  PG  PG   I SP  MAXIMUM TB COLUMN  DT  PG  STRT LIN SL RH .......SUBFILE........ C E A S R S N      C  SEQUENCE   LIST
 TY RPT S LPI WID HGT  M FRM LPP PGS TO HEADING POS POS PAGE NUM UB PN ENT    NAME   BLK/ FMT H F G T S U D      O  73 TO 80    SEQ
 MP     P             G           LT TYP POS        NUM      ML TD REC          BUF      C C S L T M S      N
--------------------------------------------------------------------------------------------------------------------------------
(E1)                                (F)                                                                      (ASLGEN'D)    18
--------------------------------------------------------------------------------------------------------------------------------
        SP  Q              E N S D C S T C C M M A G D Q PCT/RAT     P              PARTIAL   SEQUENCE   LIST
STMT SEQ BF  L  FIELD   SEG- N P E E T U O U N A I V M E L FIELD   RA- C     PICTURE     STRT NO  73 TO 80    SEQ
TYPE NO. COL F  NAME     LVL D R Q S L B T M T X N G R C F NAME    TIO T      EDIT       CHR CHR
--------------------------------------------------------------------------------------------------------------------------------
(R1)          (CUSTNO  ) 001-1                                                                               (ASLGEN'D)    19
(R1)          (ORDERNO ) 002-2                                                                               (ASLGEN'D)    20
(R1)          (ORDRDATE) 002-2                                                                               (ASLGEN'D)    21
(R1)          (ORPERSON) 002-2                                                                               (ASLGEN'D)    22
(R1)          (ORDPONUM) 002-2                                                                               (ASLGEN'D)    23
(R1)          (ORDDUDAT) 002-2                                                                               (ASLGEN'D)    24
(R1)          (ORDINVGN) 002-2                                                                               (ASLGEN'D)    25
(R1)          (ORDCMPLT) 002-2                                                                               (ASLGEN'D)    26
(R1)          (SEG30CNT) 002-2                                                                               (ASLGEN'D)    27
(R1)          (SEG50CNT) 002-2                                                                               (ASLGEN'D)    28
--------------------------------------------------------------------------------------------------------------------------------
STMT  SEQ                                                                                                     SEQUENCE   LIST
TYPE  NO.                                        TEXT                                                         73 TO 80    SEQ
--------------------------------------------------------------------------------------------------------------------------------
(T1) (101)              (BUILD THE CUSTOMER MASTER FILE#                          )                           (ASLGEN'D)    29
(T1) (102)              (LIST THE ORDER SEGMENT DATA - FIELD NAMES AS HEADINGS#   )                           (ASLGEN'D)    30
NOV 03, 2001  13.20.21                                                                                          PAGE     5
                                                  ******************************
                                                  *  PROC NAME - REPT   3      *
                                                  *  INPUT STREAM PROCEDURE    *
                                                  ******************************
--------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                          MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID    ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM       73 TO 80    SEQ
--------------------------------------------------------------------------------------------------------------------------------
(ER) (TODAY )                                                          (N)                                    (ASLGEN'D)    31
    REPORT CUSTNO ORDERNO,
           SHIPNO SHIPDATE SHIPPER FRTCOST INVNO INVDATE INAMT PAYDATE,
           CHECKNO PAIDDATE SEG40CNT
```

Figure D-2 R06PREPA (Page 17 of 26)

```
      FORMAT HEADINGS NAME
      TITLE 'BUILD THE CUSTOMER MASTER FILE'
      TITLE 'LIST THE SHIPINV SEGMENT DATA - FIELD NAMES AS HEADINGS'
    END REPORT
-------------------------------------------------------------------------------------------------------------------------------
 ST SUM V  8  PG  PG   I SP  MAXIMUM TB COLUMN  DT  PG  STRT LIN SL RH .......SUBFILE........ C E A S R S N      C  SEQUENCE   LIST
 TY RPT S LPI WID HGT  M FRM LPP PGS TO HEADING POS POS PAGE NUM UB PN ENT    NAME   BLK/ FMT H F G T S U D      O  73 TO 80    SEQ
 MP     P           G           LT TYP POS         NUM      ML TD REC           BUF      C C S L T M S      N
-------------------------------------------------------------------------------------------------------------------------------
(E1)                              (F)                                                                        (ASLGEN'D)    32
-------------------------------------------------------------------------------------------------------------------------------
        SP  Q              E N S  D  C  S  T  C  C  M  M  A  G  D Q PCT/RAT     P               PARTIAL   SEQUENCE   LIST
STMT SEQ BF  L  FIELD   SEG-  N P E  E  T  U  O  U  N  A  I  V  M  R E L  FIELD   RA- C      PICTURE     STRT NO  73 TO 80    SEQ
TYPE NO. COL F  NAME      LVL D R Q  S  L  B  T  M  T  X  N  G  R  C F  NAME    TIO T       EDIT       CHR CHR
-------------------------------------------------------------------------------------------------------------------------------
(R1)          (CUSTNO  ) 001-1                                                                            (ASLGEN'D)    33
(R1)          (ORDERNO ) 002-2                                                                            (ASLGEN'D)    34
(R1)          (SHIPNO  ) 003-3                                                                            (ASLGEN'D)    35
(R1)          (SHIPDATE) 003-3                                                                            (ASLGEN'D)    36
(R1)          (SHIPPER ) 003-3                                                                            (ASLGEN'D)    37
(R1)          (FRTCOST ) 003-3                                                                            (ASLGEN'D)    38
(R1)          (INVNO   ) 003-3                                                                            (ASLGEN'D)    39
(R1)          (INVDATE ) 003-3                                                                            (ASLGEN'D)    40
(R1)          (INAMT   ) 003-3                                                                            (ASLGEN'D)    41
(R1)          (PAYDATE ) 003-3                                                                            (ASLGEN'D)    42
(R1)          (CHECKNO ) 003-3                                                                            (ASLGEN'D)    43
(R1)          (PAIDDATE) 003-3                                                                            (ASLGEN'D)    44
(R1)          (SEG40CNT) 003-3                                                                            (ASLGEN'D)    45
-------------------------------------------------------------------------------------------------------------------------------
STMT  SEQ                                                                                                  SEQUENCE   LIST
TYPE  NO.                                          TEXT                                                    73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(T1) (101)              (BUILD THE CUSTOMER MASTER FILE#                          )                        (ASLGEN'D)    46
(T1) (102)              (LIST THE SHIPINV SEGMENT DATA - FIELD NAMES AS HEADINGS#  )                        (ASLGEN'D)    47
NOV 03, 2001  13.20.21                                                                                        PAGE     6
                                          ******************************
                                          *  PROC NAME - REPT__4     *
                                          *  INPUT STREAM PROCEDURE  *
                                          ******************************
-------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                          MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB      SEQUENCE   LIST
TYPE  DATE        REQUESTOR ID      ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM      73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(ER) (TODAY )                                                          (N)                                 (ASLGEN'D)    48
     REPORT CUSTNO ORDERNO SHIPNO,
            ITEMSHIP QTYSHIP
       FORMAT HEADINGS NAME
       TITLE 'BUILD THE CUSTOMER MASTER FILE'
       TITLE 'LIST THE ITEMSHIP SEGMENT DATA - FIELD NAMES AS HEADINGS'
     END REPORT
-------------------------------------------------------------------------------------------------------------------------------
 ST SUM V  8  PG  PG   I SP  MAXIMUM TB COLUMN  DT  PG  STRT LIN SL RH .......SUBFILE........ C E A S R S N      C  SEQUENCE   LIST
 TY RPT S LPI WID HGT  M FRM LPP PGS TO HEADING POS POS PAGE NUM UB PN ENT    NAME   BLK/ FMT H F G T S U D      O  73 TO 80    SEQ
 MP     P           G           LT TYP POS         NUM      ML TD REC           BUF      C C S L T M S      N
-------------------------------------------------------------------------------------------------------------------------------
(E1)                              (F)                                                                        (ASLGEN'D)    49
-------------------------------------------------------------------------------------------------------------------------------
        SP  Q              E N S  D  C  S  T  C  C  M  M  A  G  D Q PCT/RAT     P               PARTIAL   SEQUENCE   LIST
STMT SEQ BF  L  FIELD   SEG-  N P E  E  T  U  O  U  N  A  I  V  M  E L  FIELD   RA- C      PICTURE     STRT NO  73 TO 80    SEQ
TYPE NO. COL F  NAME      LVL D R Q  S  L  B  T  M  T  X  N  G  R  C F  NAME    TIO T       EDIT       CHR CHR
-------------------------------------------------------------------------------------------------------------------------------
(R1)          (CUSTNO  ) 001-1                                                                            (ASLGEN'D)    50
(R1)          (ORDERNO ) 002-2                                                                            (ASLGEN'D)    51
(R1)          (SHIPNO  ) 003-3                                                                            (ASLGEN'D)    52
(R1)          (ITEMSHIP) 004-4                                                                            (ASLGEN'D)    53
(R1)          (QTYSHIP ) 004-4                                                                            (ASLGEN'D)    54
-------------------------------------------------------------------------------------------------------------------------------
STMT  SEQ                                                                                                  SEQUENCE   LIST
TYPE  NO.                                          TEXT                                                    73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(T1) (101)              (BUILD THE CUSTOMER MASTER FILE#                          )                        (ASLGEN'D)    55
(T1) (102)              (LIST THE ITEMSHIP SEGMENT DATA - FIELD NAMES AS HEADINGS#  )                       (ASLGEN'D)    56
NOV 03, 2001  13.20.21                                                                                        PAGE     7
                                          ******************************
                                          *  PROC NAME - REPT__5     *
                                          *  INPUT STREAM PROCEDURE  *
                                          ******************************
-------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                          MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB      SEQUENCE   LIST
TYPE  DATE        REQUESTOR ID      ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM      73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(ER) (TODAY )                                                          (N)                                 (ASLGEN'D)    57
     REPORT CUSTNO ORDERNO,
            ITEMORD ITMPRICE ITMQTYOR QTYBKORD
       FORMAT HEADINGS NAME
       TITLE 'BUILD THE CUSTOMER MASTER FILE'
```

Figure D-2 R06PREPA (Page 18 of 26)

```
     TITLE 'LIST THE ITEMORD SEGMENT DATA - FIELD NAMES AS HEADINGS'
    END REPORT
--------------------------------------------------------------------------------------------------------------------------------
 ST SUM V  8  PG  PG   I SP  MAXIMUM TB COLUMN  DT  PG  STRT LIN SL RH .......SUBFILE..... C E A S R S N      C  SEQUENCE   LIST
 TY RPT S LPI WID HGT  M FRM LPP PGS TO HEADING POS POS PAGE NUM UB PN ENT    NAME   BLK/ FMT H F G T S U D      O  73 TO 80    SEQ
 MP     P            G           LT TYP POS         NUM      ML TD REC          BUF      C C S L T M S      N
--------------------------------------------------------------------------------------------------------------------------------
(E1)                                (F)                                                                        (ASLGEN'D)    58
--------------------------------------------------------------------------------------------------------------------------------
        SP  Q              E N S D C S T C C M M A G D Q PCT/RAT     P                PARTIAL   SEQUENCE   LIST
STMT SEQ  BF  L  FIELD  SEG-  N P E E T U O U N A I V M E L  FIELD   RA- C     PICTURE     STRT NO  73 TO 80    SEQ
TYPE NO.  COL F  NAME     LVL  D R Q S L B T M T X N G R C F  NAME    TIO T      EDIT      CHR CHR
--------------------------------------------------------------------------------------------------------------------------------
(R1)          (CUSTNO  ) 001-1                                                                                 (ASLGEN'D)    59
(R1)          (ORDERNO ) 002-2                                                                                 (ASLGEN'D)    60
(R1)          (ITEMORD ) 005-3                                                                                 (ASLGEN'D)    61
(R1)          (ITMPRICE) 005-3                                                                                 (ASLGEN'D)    62
(R1)          (ITMQTYOR) 005-3                                                                                 (ASLGEN'D)    63
(R1)          (QTYBKORD) 005-3                                                                                 (ASLGEN'D)    64
--------------------------------------------------------------------------------------------------------------------------------
STMT  SEQ                                                                                                       SEQUENCE   LIST
TYPE  NO.                                          TEXT                                                         73 TO 80    SEQ
--------------------------------------------------------------------------------------------------------------------------------
(T1) (101)              (BUILD THE CUSTOMER MASTER FILE#                          )                            (ASLGEN'D)    65
(T1) (102)              (LIST THE ITEMORD SEGMENT DATA - FIELD NAMES AS HEADINGS#  )                           (ASLGEN'D)    66
NOV 03, 2001  13.20.21                                                                                           PAGE     8
                                          ******************************
                                          *  PROC NAME - REPT___6     *
                                          *  INPUT STREAM PROCEDURE   *
                                          ******************************
--------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                          MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE         REQUESTOR ID     ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM       73 TO 80    SEQ
--------------------------------------------------------------------------------------------------------------------------------
(ER) (TODAY )                                                                (N)                                (ASLGEN'D)    67
    REPORT CUSTNO,
           INSTNO INSTCONT INSTOPS INSTOSRL INPRODID
      FORMAT HEADINGS NAME
      TITLE 'BUILD THE CUSTOMER MASTER FILE'
      TITLE 'LIST THE INSTALL SEGMENT DATA - FIELD NAMES AS HEADINGS'
    END REPORT
--------------------------------------------------------------------------------------------------------------------------------
 ST SUM V  8  PG  PG   I SP  MAXIMUM TB COLUMN  DT  PG  STRT LIN SL RH .......SUBFILE..... C E A S R S N      C  SEQUENCE   LIST
 TY RPT S LPI WID HGT  M FRM LPP PGS TO HEADING POS POS PAGE NUM UB PN ENT    NAME   BLK/ FMT H F G T S U D      O  73 TO 80    SEQ
 MP     P            G           LT TYP POS         NUM      ML TD REC          BUF      C C S L T M S      N
--------------------------------------------------------------------------------------------------------------------------------
(E1)                                (F)                                                                        (ASLGEN'D)    68
--------------------------------------------------------------------------------------------------------------------------------
        SP  Q              E N S D C S T C C M M A G D Q PCT/RAT     P                PARTIAL   SEQUENCE   LIST
STMT SEQ  BF  L  FIELD  SEG-  N P E E T U O U N A I V M E L  FIELD   RA- C     PICTURE     STRT NO  73 TO 80    SEQ
TYPE NO.  COL F  NAME     LVL  D R Q S L B T M T X N G R C F  NAME    TIO T      EDIT      CHR CHR
--------------------------------------------------------------------------------------------------------------------------------
(R1)          (CUSTNO  ) 001-1                                                                                 (ASLGEN'D)    69
(R1)          (INSTNO  ) 006-2                                                                                 (ASLGEN'D)    70
(R1)          (INSTCONT) 006-2                                                                                 (ASLGEN'D)    71
(R1)          (INSTOPS ) 006-2                                                                                 (ASLGEN'D)    72
(R1)          (INSTOSRL) 006-2                                                                                 (ASLGEN'D)    73
(R1)          (INPRODID) 006-2                                                                                 (ASLGEN'D)    74
--------------------------------------------------------------------------------------------------------------------------------
STMT  SEQ                                                                                                       SEQUENCE   LIST
TYPE  NO.                                          TEXT                                                         73 TO 80    SEQ
--------------------------------------------------------------------------------------------------------------------------------
(T1) (101)              (BUILD THE CUSTOMER MASTER FILE#                          )                            (ASLGEN'D)    75
(T1) (102)              (LIST THE INSTALL SEGMENT DATA - FIELD NAMES AS HEADINGS#  )                           (ASLGEN'D)    76
NOV 03, 2001  13.20.21                                                                                           PAGE     9
** MK4W204  TYPE 0  NUMBER OF MESSAGES PRINTED IS    0.                                                                  *****
                    M4NEW    -       5 RECORDS OUTPUT
                    M4TRAN   -      35 RECORDS INPUT
                    M4REPO   -     106 RECORDS OUTPUT
                    M4INPUT  -      41 RECORDS INPUT
                    M4LIST   -     331 RECORDS OUTPUT
                      15 TRACKS ASSIGNED TO M4LIB    --   13 TRACKS NOT FULL. LIBRARY DIRECTORY BLOCKING FACTOR IS  800.
** MK4W209  TYPE 0  1010376 BYTES OF MAIN STORAGE UNUSED DURING DECODING PHASE                                        *****
                    1002946 BYTES OF MAIN STORAGE UNUSED DURING COMPILATION PHASE
                     564888 BYTES OF MAIN STORAGE UNUSED DURING PROCESSING PHASE
** MK4ED02  TYPE 0  REQUEST-REPT___1  REPORT-1  REQUESTOR ID-                                                        *****
11/03/01                                    BUILD THE CUSTOMER MASTER FILE                                    PAGE 1
                                       LIST THE CUSTOMER SEGMENT DATA - FIELD NAMES AS HEADINGS
--------------------------------------------------------------------------------------------------------------------------------
  CUSTNO           CUSTNAME              CUSTPH          CUSTADDR                CITY            STATE  CUSTZIP  SEG60CNT
--------------------------------------------------------------------------------------------------------------------------------
  00002   FINANCIAL SERVICES           2125551234  4321 MAIN ST            ALBANY               NY     10111        1
  00003   THE INSURANCE COMPANY        1235557777  323 GREENVIEW ROAD      ST LOUIS             MI     22334        1
  00005   RECREATIONAL TOYS            8185555131  3151 EXTREME RIDGE DRIVE WINNITKA            CA     91306        1
  00010   HOME IMPROVEMENT SUPPLIES    3435553434  987 LUMBERMAN TRAIL     WOODLAND PLAINS      OR     34555        1
  00015   CLOTHES LINE INC             7655559876  777 PLAID BLVD          COTTON LAKE          NC     87552        1
```

Figure D-2 R06PREPA (Page 19 of 26)

```
** MK4FT03  TYPE 0  END OF REPORT.                                                                                    *****
** MK4ED02  TYPE 0  REQUEST-REPT___2  REPORT-1  REQUESTOR ID-                                                        *****
11/03/01                                  BUILD THE CUSTOMER MASTER FILE                                  PAGE 1
                                       LIST THE ORDER SEGMENT DATA - FIELD NAMES AS HEADINGS
-------------------------------------------------------------------------------------------------------------------------
  CUSTNO  ORDERNO  ORDRDATE  ORPERSON  ORDPONUM  ORDDUDAT  ORDINVGN  ORDCMPLT  SEG30CNT  SEG50CNT
-------------------------------------------------------------------------------------------------------------------------
  00002    20103    033101     JJK      P3151     042801      Y         Y          1         1
  00002    20105    091501     MBK      P5131     092801      Y         Y          1         1
  00003    30101    052301     JCR      P2355     061501      Y         Y          1         1
  00005      -        -        -         -         -         -         -          -         -
  00010    01101    011301     GOA      P0431     013101      Y         Y          1         1
  00010    01104    021501     JJK      P0528     022801      Y         Y          1         1
  00015      -        -        -         -         -         -         -          -         -
** MK4FT06  TYPE 1       18 FIELDS WERE PRINTED AS INVALID, MISSING OR OVERFLOW.                                    *****
** MK4FT03  TYPE 0  END OF REPORT.                                                                                   *****
** MK4ED02  TYPE 0  REQUEST-REPT___3  REPORT-1  REQUESTOR ID-                                                        *****
11/03/01                                  BUILD THE CUSTOMER MASTER FILE                                  PAGE 1
                                     LIST THE SHIPINV SEGMENT DATA - FIELD NAMES AS HEADINGS
-------------------------------------------------------------------------------------------------------------------------
  CUSTNO  ORDERNO  SHIPNO  SHIPDATE  SHIPPER   FRTCOST    INVNO  INVDATE    INAMT     PAYDATE  CHECKNO  PAIDDATE  SEG40CNT
-------------------------------------------------------------------------------------------------------------------------
  00002    20103    D25C    042501     UPS        $3.38  UD25A  042501     $34.38    052501   C032599   060101        1
  00002    20105    I15J    091501     UPS         4.43  UI15J  091801      39.43    101801   C051290   111101        1
  00003    30101    E28X    052801     POF          .98  PE28X  052801       5.48    062801   C000123   062901        1
  00005      -        -       -         -           -     -      -           -        -        -         -           -
  00010    01101    A25G    012501     HAN          .00  HA25G  012501     475.00    012501   C005533   012501        1
  00010    01104    B19K    021901     HAN          .00  HB19K  021901     535.00    021901   C005601   021901        1
  00015      -        -       -         -           -     -      -           -        -        -         -           -
** MK4FT06  TYPE 1       24 FIELDS WERE PRINTED AS INVALID, MISSING OR OVERFLOW.                                    *****
** MK4FT03  TYPE 0  END OF REPORT.                                                                                   *****
** MK4ED02  TYPE 0  REQUEST-REPT___4  REPORT-1  REQUESTOR ID-                                                        *****
11/03/01                                  BUILD THE CUSTOMER MASTER FILE                                  PAGE 1
                                     LIST THE ITEMSHIP SEGMENT DATA - FIELD NAMES AS HEADINGS
-------------------------------------------------------------------------------------------------------------------------
  CUSTNO  ORDERNO  SHIPNO  ITEMSHIP   QTYSHIP
-------------------------------------------------------------------------------------------------------------------------
  00002    20103    D25C   BL-H11A          3
  00002    20105    I15J   BL-H11B          3
  00003    30101    E28X   BL-B01B          5
  00005      -        -       -             -
  00010    01101    A25G   BL-T01B          1
  00010    01104    B19K   BL-T02B          1
  00015      -        -       -             -
** MK4FT06  TYPE 1        8 FIELDS WERE PRINTED AS INVALID, MISSING OR OVERFLOW.                                    *****
** MK4FT03  TYPE 0  END OF REPORT.                                                                                   *****
** MK4ED02  TYPE 0  REQUEST-REPT___5  REPORT-1  REQUESTOR ID-                                                        *****
11/03/01                                  BUILD THE CUSTOMER MASTER FILE                                  PAGE 1
                                     LIST THE ITEMORD SEGMENT DATA - FIELD NAMES AS HEADINGS
-------------------------------------------------------------------------------------------------------------------------
  CUSTNO  ORDERNO  ITEMORD  ITMPRICE    ITMQTYOR    QTYBKORD
-------------------------------------------------------------------------------------------------------------------------
  00002    20103   BL-H11A    $31.00          3
  00002    20105   BL-H11B     35.00          3
  00003    30101   BL-B01B      4.50          5
  00005      -       -          -             -          -
  00010    01101   BL-T01B    475.00          1
  00010    01104   BL-T02B    535.00          1
  00015      -       -          -             -          -
** MK4FT06  TYPE 1       10 FIELDS WERE PRINTED AS INVALID, MISSING OR OVERFLOW.                                    *****
** MK4FT03  TYPE 0  END OF REPORT.                                                                                   *****
** MK4ED02  TYPE 0  REQUEST-REPT___6  REPORT-1  REQUESTOR ID-                                                        *****
11/03/01                                  BUILD THE CUSTOMER MASTER FILE                                  PAGE 1
                                     LIST THE INSTALL SEGMENT DATA - FIELD NAMES AS HEADINGS
-------------------------------------------------------------------------------------------------------------------------
  CUSTNO  INSTNO  INSTCONT  INSTOPS   INSTOSRL  INPRODID
-------------------------------------------------------------------------------------------------------------------------
  00002    A031     JJK     OS/390     2.08      13.8
  00003    B051     JCR     VSE/ESA    2.5       13.7
  00005    C513     MBK     VM/CMS     3.10      13.0
  00010    D432     GOA     OS/390     2.10      3.1
  00015    E919     JJK     OS/390     2.11      14.0
** MK4FT03  TYPE 0  END OF REPORT.                                                                                   *****
** MK4W204  TYPE 0  NUMBER OF MESSAGES PRINTED IS   16.                                                             *****
                    M4REPO   -      105 RECORDS INPUT
                    M4LIST   -       86 RECORDS OUTPUT
** MK4W209  TYPE 0   996002 BYTES OF MAIN STORAGE UNUSED DURING REPORTING PHASE                                      *****

NOV 03, 2001  13.20.24                                                                                    PAGE      1

                                          ********************************************
                                          * VISION:Builder   4400  (OS/390 - 14.0)  *
                                          *          COPYRIGHT (C) 2002              *
                                          * COMPUTER ASSOCIATES INTERNATIONAL, INC. *
                                          ********************************************
VISION:Builder JCR1 14.0 110201              ENABLED FOR IBM LANGUAGE ENVIRONMENT              BUILD STAMP = 101306,12:51:24.
                                          COMLIB  MODIFICATIONS (PTFs, APARs) INSTALLED
```

Figure D-2 R06PREPA (Page 20 of 26)

```
                                  201,202,203,204,205,206,207,208,209,210,212,213,214,215,216
=                                                                                                                                =
=                                  I N S T A L L A T I O N   P A R A M E T E R S   ( M 4 P A R A M S , M A R K L I B P )         =
=                                                                                                                                =
=  SYSTEM DELIMITER:   #          PAGE HEIGHT:        66         M4LIST WIDTH:       132        DEF WIDTH OF PAGE:  0          =
=                                                                                                                                =
=  AUTO GRAND:         N          HEADING CHAR:       -          SUBTITLE REPEAT:    N          INVALID FIELD:      *          =
=                                                                                                                                =
=  MISSING FIELD:      -          NON-EDIT FIELD:     +          PERCENT CHAR:       %          LEFT SEPARATOR:     (          =
=                                                                                                                                =
=  RIGHT SEPARATOR:    )          SINGLE SEPARATOR:   ,          SOURCE SPACING:     1          PRINT MESSAGES:     Y          =
=                                                                                                                                =
=  CONSOLE MESSAGES:   N          M4REPO BLOCKSIZE:   4,096      INPUT I/O BUFFERS:  2          OUTPUT I/O BUFFERS: 1          =
=                                                                                                                                =
=  SNGL-STEP STORAGE:  8,192      SNGL-STEP SORTSIZE: 524,288    DIGIT SELECT CHAR:  9          ZERO SUPPRESS CHAR: Z          =
=                                                                                                                                =
=  CURRENCY CHAR:      $          PLUS CHAR:          +          MINUS CHAR:         -          CHECK PROTECT CHAR: *          =
=                                                                                                                                =
=  DECIMAL CHAR:       .          GROUPING CHAR:      ,          PRIMARY PLOT CHAR:  X          SECONDARY PLOT CHAR:*          =
=                                                                                                                                =
=  FIT PLOT CHAR:      .          HORIZONTAL AXIS:    _          HORIZONTAL HASH:    |          VERTICAL AXIS:     |          =
=                                                                                                                                =
=  VERTICAL HASH:      -          MINUTES/HOUR:       60         SECONDS/MINUTE:     60         TIME DELIMITER:     HH:MM:SS   =
=                                                                                                                                =
=  DATE FORMAT:      MMM DD, YYYY TODAY FORMAT+DELIM: MM/DD/YY   ISDATE DELIMITER:   YYYY-MM-DD JULIAN DELIMITER:   YY.DDD    =
=                                                                                                                                =
=  SORT PROGRAM CODE:  2          MINCORE VALUE:      12 K       ALT M4LIST WIDTH:   264        ALT DEF W/OF PAGE:  0          =
=                                                                                                                                =
=  MAX LINES OF TRACE: 1,024      ITEM TRACKING:      0          SUPPRESS NDS REPT?: N          DEFAULT MAXGETMN:   1,024 K    =
=                                                                                                                                =
=  CONDITION CODE 1:   0          CONDITION CODE 2:   4          CONDITION CODE 3:   0          CONDITION CODE 4:   16         =
=                                                                                                                                =

NOV 03, 2001  13.20.24                                                                                                PAGE      2

   RUN     STMT    FILE    O N S D U T R A B S D S L  RB  R S M R S            START           END             SEQUENCE    LIST
   NAME    TYPE    NAME    L E E I P R E U U R L C S  PL  J O O F S            SEARCH          SEARCH          73 TO 80    SEQ
                           D W Q R D N P D F T M N T  TK  T P P O R

(BLDCUSTH)  (RC)  (CUSTHIST) (S)          (S)        (T,G)          (R)                                    (ASLGEN'D)      1
     CONTROL NAME BLDCUSTH TERM LISTGEN
     FILE MASTER INPUT  CUSTHIST
     FILE SUBF1  CUSTHST
     FILE REPORT

   RUN     STMT  LOGICAL  LBL  DTF/DD      KEY      KEY      KEY      QUALIFIER   USER I/O  ARRAY       PASSWORD/   SEQUENCE    LIST
   NAME    TYPE  FILENAME       NAME      FIELD-1  FIELD-2  FIELD-3   ICF CRD DBI   MODULE   QUAL        AUTH.-ID    73 TO 80    SEQ

(BLDCUSTH)  (RF)  (CUSTHIST) (S) (M4OLD    )                                                              (ASLGEN'D)      2
(BLDCUSTH)  (RF)  (CUSTHST ) (S) (M4SUBF1 )                                                              (ASLGEN'D)      3


=                                                                                                                                =
=                                                  F I L E   S U M M A R Y                                                       =
=                                                                                                                                =
= JOBNAME: KREJO04X                          STEPNAME: PROC2                                                                     =
=                                                                                                                                =
=  FILE NAME: M4OLD                          DSNAME: KREJO04.KREJO04X.JOB04981.D0000105.?                                        =
=  FILE NAME: M4SUBF1                        DSNAME: BUILDER.R140.CUSTOMER.HISTORY.R06#1                                         =
=  FILE NAME: M4LIB                          DSNAME: BUILDER.R140.M4LIB.R06#1                                                    =
=                                                                                                                                =

NOV 03, 2001  13.20.24                                                                                                PAGE      3

                                                  ****************************
                                                  *  PROC NAME - REPT__1     *
                                                  *  INPUT STREAM PROCEDURE  *
                                                  ****************************

STMT REPORT                                MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB       SEQUENCE    LIST
TYPE  DATE          REQUESTOR ID          ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM       73 TO 80    SEQ

(ER) (TODAY )                                                                    (N)                                 (ASLGEN'D)      4
     REPORT CUSTNO ORDERNO ORDRDATE ORPERSON ORDPONUM
       FORMAT HEADINGS NAME
       TITLE 'BUILD THE CUSTOMER HISTORY FILE'
       TITLE 'LIST THE DATA - FIELD NAMES AS HEADINGS'
     END REPORT

 ST SUM V  8  PG  PG   I SP  MAXIMUM TB COLUMN  DT  PG  STRT LIN SL RH .......SUBFILE..... C E A S R S N      C  SEQUENCE    LIST
 TY RPT S LPI WID HGT  M FRM LPP PGS TO HEADING POS POS PAGE NUM UB PN ENT    NAME   BLK/ FMT H F G T S U D      O  73 TO 80    SEQ
 MP     P             G            LT TYP POS         NUM      ML TD REC          BUF      C C S L T M S      N

(E1)                                (F)                                                                      (ASLGEN'D)      5
```

Figure D-2 R06PREPA (Page 21 of 26)

```
------------------------------------------------------------------------------------------------------------------------------------
         SP  Q                E  N  S  D  C  S  T  C  C  M  M  A  G  D  Q  PCT/RAT       P                    PARTIAL   SEQUENCE   LIST
STMT SEQ  BF  L  FIELD    SEG-  N  P  E  E  T  U  O  U  N  A  I  V  M  E  L  FIELD    RA- C       PICTURE      STRT NO   73 TO 80    SEQ
TYPE NO.   COL F  NAME       LVL  D  R  Q  S  L  B  T  M  T  X  N  G  R  C  F  NAME     TIO T        EDIT       CHR CHR
------------------------------------------------------------------------------------------------------------------------------------
(R1)          (CUSTNO  ) 001-1                                                                                      (ASLGEN'D)      6
(R1)          (ORDERNO ) 001-1                                                                                      (ASLGEN'D)      7
(R1)          (ORDRDATE) 001-1                                                                                      (ASLGEN'D)      8
(R1)          (ORPERSON) 001-1                                                                                      (ASLGEN'D)      9
(R1)          (ORDPONUM) 001-1                                                                                      (ASLGEN'D)     10
------------------------------------------------------------------------------------------------------------------------------------
STMT  SEQ                                                                                                            SEQUENCE   LIST
TYPE  NO.                                        TEXT                                                                73 TO 80    SEQ
------------------------------------------------------------------------------------------------------------------------------------
(T1)  (101)                  (BUILD THE CUSTOMER HISTORY FILE#                              )                       (ASLGEN'D)     11
(T1)  (102)                  (LIST THE DATA - FIELD NAMES AS HEADINGS#                      )                       (ASLGEN'D)     12
NOV 03, 2001  13.20.24                                                                                                   PAGE      4
                                             *****************************
                                             *  PROC NAME - SUBF   2      *
                                             *  INPUT STREAM PROCEDURE    *
                                             *****************************
------------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                                MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ         BACK BR RNT AB         SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID          ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM        73 TO 80    SEQ
------------------------------------------------------------------------------------------------------------------------------------
(ER)                                                                           (N)                                  (ASLGEN'D)     13
     EXTRACT FILE CUSTHST ENTIRE N RECFM FIXED
------------------------------------------------------------------------------------------------------------------------------------
 ST SUM V  8  PG  PG   I SP  MAXIMUM TB COLUMN  DT  PG  STRT LIN SL RH .......SUBFILE........ C E A S R S N      C  SEQUENCE   LIST
 TY RPT S LPI WID HGT  M FRM LPP PGS TO HEADING POS POS PAGE NUM UB PN ENT     NAME   BLK/ FMT H F G T S U D      O  73 TO 80    SEQ
 MP     P              G             LT TYP POS          NUM      ML TD REC             BUF      C C S L T M S      N
------------------------------------------------------------------------------------------------------------------------------------
(E1)                                                         (NR) (N) (CUSTHST )       (F)                          (ASLGEN'D)     14
NOV 03, 2001  13.20.24                                                                                                   PAGE      5
** MK4W204  TYPE 0  NUMBER OF MESSAGES PRINTED IS    0.                                                                        *****
                    M4OLD   -        9 RECORDS INPUT
                    M4REPO  -       22 RECORDS OUTPUT
                    M4SUBF1 -        9 RECORDS OUTPUT
                    M4INPUT -       10 RECORDS INPUT
                    M4LIST  -      133 RECORDS OUTPUT
                      15 TRACKS ASSIGNED TO M4LIB   --   13 TRACKS NOT FULL. LIBRARY DIRECTORY BLOCKING FACTOR IS  800.
** MK4W209  TYPE 0  1017928 BYTES OF MAIN STORAGE UNUSED DURING DECODING PHASE                                                 *****
                    1019160 BYTES OF MAIN STORAGE UNUSED DURING COMPILATION PHASE
                     565248 BYTES OF MAIN STORAGE UNUSED DURING PROCESSING PHASE
** MK4ED02  TYPE 0  REQUEST-REPT___1  REPORT-1  REQUESTOR ID-                                                                  *****
11/03/01                                   BUILD THE CUSTOMER HISTORY FILE                                             PAGE 1
                                       LIST THE DATA - FIELD NAMES AS HEADINGS
------------------------------------------------------------------------------------------------------------------------------------
  CUSTNO  ORDERNO  ORDRDATE  ORPERSON  ORDPONUM
------------------------------------------------------------------------------------------------------------------------------------
  00001    10001    031500     JJK      P0151
  00001    10004    051500     JJK      P0551
  00001    10005    071500     JJK      P0751
  00002    20003    043000     JJK      P4301
  00002    20005    082300     MBK      P8231
  00003    30001    011100     JCR      P1111
  00010    01001    051300     GOA      P0531
  00010    01004    072500     JJK      P0728
  00020    02003    091900     JJK      P0919
** MK4FT03  TYPE 0  END OF REPORT.                                                                                             *****
** MK4W204  TYPE 0  NUMBER OF MESSAGES PRINTED IS    2.                                                                        *****
                    M4REPO  -       21 RECORDS INPUT
                    M4LIST  -       18 RECORDS OUTPUT
** MK4W209  TYPE 0  1005568 BYTES OF MAIN STORAGE UNUSED DURING REPORTING PHASE                                                *****

NOV 03, 2001  13.20.26                                                                                                   PAGE      1
                                       *********************************************
                                       *  VISION:Builder   4400  (OS/390 - 14.0)   *
                                       *            COPYRIGHT (C) 2002             *
                                       *  COMPUTER ASSOCIATES INTERNATIONAL, INC.  *
                                       *********************************************
VISION:Builder JCR1 14.0 110201            ENABLED FOR IBM LANGUAGE ENVIRONMENT                 BUILD STAMP = 101306,12:51:24.
                                          COMLIB  MODIFICATIONS (PTFs, APARs) INSTALLED
                                 201,202,203,204,205,206,207,208,209,210,212,213,214,215,216
====================================================================================================================================
=                                                                                                                                  =
=                    I N S T A L L A T I O N   P A R A M E T E R S   ( M 4 P A R A M S , M A R K L I B P )                        =
=                                                                                                                                  =
=  SYSTEM DELIMITER:   #          PAGE HEIGHT:        66          M4LIST WIDTH:       132          DEF WIDTH OF PAGE:  0          =
=                                                                                                                                  =
=  AUTO GRAND:         N          HEADING CHAR:       -           SUBTITLE REPEAT:    N            INVALID FIELD:      *          =
=                                                                                                                                  =
=  MISSING FIELD:      -          NON-EDIT FIELD:     +           PERCENT CHAR:       %            LEFT SEPARATOR:     (          =
=                                                                                                                                  =
=  RIGHT SEPARATOR:    )          SINGLE SEPARATOR:   ,           SOURCE SPACING:     1            PRINT MESSAGES:     Y          =
=                                                                                                                                  =
```

Figure D-2        R06PREPA  (Page 22 of 26)

```
=  CONSOLE MESSAGES:   N          M4REPO BLOCKSIZE:   4,096     INPUT I/O BUFFERS:  2          OUTPUT I/O BUFFERS: 1          =
=                                                                                                                               =
=  SNGL-STEP STORAGE:  8,192      SNGL-STEP SORTSIZE: 524,288   DIGIT SELECT CHAR:  9          ZERO SUPPRESS CHAR: Z          =
=                                                                                                                               =
=  CURRENCY CHAR:      $          PLUS CHAR:          +         MINUS CHAR:         -          CHECK PROTECT CHAR: *          =
=                                                                                                                               =
=  DECIMAL CHAR:       .          GROUPING CHAR:      ,         PRIMARY PLOT CHAR:  X          SECONDARY PLOT CHAR:*          =
=                                                                                                                               =
=  FIT PLOT CHAR:      .          HORIZONTAL AXIS:    _         HORIZONTAL HASH:    |          VERTICAL AXIS:     |          =
=                                                                                                                               =
=  VERTICAL HASH:      -          MINUTES/HOUR:       60        SECONDS/MINUTE:     60         TIME DELIMITER:     HH:MM:SS   =
=                                                                                                                               =
=  DATE FORMAT:      MM DD, YYYY  TODAY FORMAT+DELIM: MM/DD/YY  ISDATE DELIMITER:   YYYY-MM-DD  JULIAN DELIMITER:   YY.DDD    =
=                                                                                                                               =
=  SORT PROGRAM CODE:  2          MINCORE VALUE:      12 K      ALT M4LIST WIDTH:   264        ALT DEF W/OF PAGE:  0          =
=                                                                                                                               =
=  MAX LINES OF TRACE: 1,024      ITEM TRACKING:      0         SUPPRESS NDS REPT?: N          DEFAULT MAXGETMN:   1,024 K    =
=                                                                                                                               =
=  CONDITION CODE 1:   0          CONDITION CODE 2:   4         CONDITION CODE 3:   0          CONDITION CODE 4:   16         =
=                                                                                                                               =
```

NOV 03, 2001  13.20.26 PAGE 2

```
   RUN     STMT    FILE    O N S D U T R A B S D S L  RB  R S M R S            START           END             SEQUENCE   LIST
   NAME    TYPE    NAME    L E E I P R E U U R L C S  PL  J O O F S            SEARCH          SEARCH          73 TO 80    SEQ
                           D W Q R D N P D F T M N T  TK  T P P O R
-------------------------------------------------------------------------------------------------------------------------------
(BLDIHIST) (RC) (ITEMHIST) (S)         (S)       (T,G)          (R)                                          (ASLGEN'D)     1
    CONTROL NAME BLDIHIST TERM LISTGEN
    FILE MASTER INPUT  ITEMHIST
    FILE SUBF1  ITEMHIST
    FILE REPORT
-------------------------------------------------------------------------------------------------------------------------------
   RUN     STMT  LOGICAL  LBL  DTF/DD      KEY       KEY       KEY      QUALIFIER   USER I/O  ARRAY       PASSWORD/  SEQUENCE   LIST
   NAME    TYPE  FILENAME       NAME      FIELD-1   FIELD-2   FIELD-3   ICF CRD DBI   MODULE   QUAL        AUTH.-ID   73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(BLDIHIST) (RF) (ITEMHIST) (S) (M4OLD    )                                                                  (ASLGEN'D)     2
(BLDIHIST) (RF) (ITEMHIST) (S) (M4SUBF1 )                                                                   (ASLGEN'D)     3
```

```
=                                                                                                                               =
=                                                  F I L E    S U M M A R Y                                                     =
=                                                                                                                               =
= JOBNAME: KREJO04X                          STEPNAME: PROC3                                                                    =
=                                                                                                                               =
=  FILE NAME: M4OLD                          DSNAME: KREJO04.KREJO04X.JOB04981.D0000107.?                                       =
=  FILE NAME: M4SUBF1                        DSNAME: BUILDER.R140.ITEM.HISTORY.R06#1                                            =
=  FILE NAME: M4LIB                          DSNAME: BUILDER.R140.M4LIB.R06#1                                                   =
=                                                                                                                               =
```

NOV 03, 2001  13.20.26 PAGE 3

```
                                        ****************************
                                        *  PROC NAME - REPT   1    *
                                        *  INPUT STREAM PROCEDURE  *
                                        ****************************
-------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                          MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ         BACK BR RNT AB      SEQUENCE   LIST
TYPE  DATE        REQUESTOR ID      ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM      73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(ER)                                                                    (N)                                  (ASLGEN'D)     4
    REPORT CUSTNO ORDERNO ITEMNO ITMPRICE ITEMQTY
      TITLE 'BUILD THE ITEM HISTORY FILE - A SEQUENTIAL DATA SET'
      TITLE 'LIST THE DATA'
    END REPORT
-------------------------------------------------------------------------------------------------------------------------------
 ST SUM V  8  PG  PG   I SP  MAXIMUM TB COLUMN  DT  PG  STRT LIN SL RH .......SUBFILE..... C E A S R S N      C  SEQUENCE   LIST
 TY RPT S LPI WID HGT  M FRM LPP PGS TO HEADING POS POS PAGE NUM UB PN ENT    NAME    BLK/ FMT H F G T S U D      O 73 TO 80    SEQ
 MP     P            G           LT TYP POS         NUM      ML TD REC          BUF     C C S L T M S      N
-------------------------------------------------------------------------------------------------------------------------------
(E1)                                                                                                         (ASLGEN'D)     5
-------------------------------------------------------------------------------------------------------------------------------
         SP  Q              E N S D C S T C C M M A G D Q PCT/RAT     P              PARTIAL   SEQUENCE   LIST
STMT SEQ BF  L  FIELD  SEG- N P E E T U O U N A I V M E L  FIELD  RA- C    PICTURE   STRT NO  73 TO 80    SEQ
TYPE NO. COL F  NAME     LVL D R Q S L B T M T X N G R C F  NAME   TIO T     EDIT      CHR CHR
-------------------------------------------------------------------------------------------------------------------------------
(R1)          (CUSTNO  ) 001-1                                                                               (ASLGEN'D)     6
(R1)          (ORDERNO ) 001-1                                                                               (ASLGEN'D)     7
(R1)          (ITEMNO  ) 001-1                                                                               (ASLGEN'D)     8
(R1)          (ITMPRICE) 001-1                                                                               (ASLGEN'D)     9
(R1)          (ITEMQTY ) 001-1                                                                               (ASLGEN'D)    10
-------------------------------------------------------------------------------------------------------------------------------
STMT  SEQ                                                                                                    SEQUENCE   LIST
TYPE  NO.                                      TEXT                                                          73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
```

Figure D-2 R06PREPA (Page 23 of 26)

```
(T1) (101)                (BUILD THE ITEM HISTORY FILE - A SEQUENTIAL DATA SET#       )                    (ASLGEN'D)    11
(T1) (102)                (LIST THE DATA#                                            )                    (ASLGEN'D)    12
NOV 03, 2001  13.20.26                                                                                      PAGE      4
                                                  ****************************
                                                  *  PROC NAME - SUBF   2    *
                                                  *  INPUT STREAM PROCEDURE  *
                                                  ****************************

-------------------------------------------------------------------------------------------------------------------------
STMT REPORT                              MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID         ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM      73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------
(ER)                                                                          (N)                                (ASLGEN'D)    13
     EXTRACT FILE ITEMHIST ENTIRE N RECFM FIXED
-------------------------------------------------------------------------------------------------------------------------
 ST SUM V  8  PG  PG   I SP  MAXIMUM TB COLUMN  DT  PG  STRT LIN SL RH .......SUBFILE........ C E A S R S N      C  SEQUENCE   LIST
 TY RPT S LPI WID HGT  M FRM LPP PGS TO HEADING POS POS PAGE NUM UB PN ENT    NAME   BLK/ FMT H F G T S U D      O  73 TO 80    SEQ
 MP     P             G            LT TYP POS         NUM      ML TD REC           BUF      C C S L T M S      N
-------------------------------------------------------------------------------------------------------------------------
(E1)                                                                 (NR) (N) (ITEMHIST)      (F)                (ASLGEN'D)    14
NOV 03, 2001  13.20.26                                                                                      PAGE      5
** MK4W204  TYPE 0  NUMBER OF MESSAGES PRINTED IS    0.                                                                *****
                    M4OLD    -       16 RECORDS INPUT
                    M4REPO   -       29 RECORDS OUTPUT
                    M4SUBF1  -       16 RECORDS OUTPUT
                    M4INPUT  -        9 RECORDS INPUT
                    M4LIST   -      132 RECORDS OUTPUT
                      15 TRACKS ASSIGNED TO M4LIB    --   13 TRACKS NOT FULL. LIBRARY DIRECTORY BLOCKING FACTOR IS  800.
** MK4W209  TYPE 0  1017928 BYTES OF MAIN STORAGE UNUSED DURING DECODING PHASE                                        *****
                    1019152 BYTES OF MAIN STORAGE UNUSED DURING COMPILATION PHASE
                     565248 BYTES OF MAIN STORAGE UNUSED DURING PROCESSING PHASE
** MK4ED02  TYPE 0  REQUEST-REPT___1  REPORT-1  REQUESTOR ID-                                                        *****
                                    BUILD THE ITEM HISTORY FILE - A SEQUENTIAL DATA SET                    PAGE 1
                                                  LIST THE DATA
-------------------------------------------------------------------------------------------------------------------------
  CUSTOMER  ORDER   ITEM     ITEM      ORDER
    NUMBER  NUMBER  NUMBER   PRICE     QUANTITY
-------------------------------------------------------------------------------------------------------------------------
    00001   10001   M55      $15.95          1
    00001   10001   N94       39.95          1
    00001   10004   M52       19.95          1
    00001   10004   O17        3.95          1
    00001   10005   O13        7.95          1
    00002   20003   H11       25.50          3
    00002   20003   H68        7.95          3
    00002   20009   E68       27.50          1
    00003   30001   E68       27.50          1
    00010   01001   N34       39.95          1
    00010   01004   E68       27.50          1
    00010   01004   M52       19.95          1
    00015   15001   O17        3.95          1
    00015   15006   N11       23.50          1
    00015   15006   N19       18.95          1
    00020   02003   E13       28.95          1
** MK4FT03  TYPE 0  END OF REPORT.                                                                                   *****
** MK4W204  TYPE 0  NUMBER OF MESSAGES PRINTED IS    2.                                                             *****
                    M4REPO   -       28 RECORDS INPUT
                    M4LIST   -       26 RECORDS OUTPUT
** MK4W209  TYPE 0  1005270 BYTES OF MAIN STORAGE UNUSED DURING REPORTING PHASE                                      *****

NOV 03, 2001  13.20.27                                                                                      PAGE      1
                                          ******************************************
                                          *  VISION:Builder   4400  (OS/390 - 14.0)  *
                                          *           COPYRIGHT (C) 2002             *
                                          *  COMPUTER ASSOCIATES INTERNATIONAL, INC. *
                                          ******************************************
VISION:Builder JCR1 14.0 110201              ENABLED FOR IBM LANGUAGE ENVIRONMENT                BUILD STAMP = 101306,12:51:24.
                                           COMLIB  MODIFICATIONS (PTFs, APARs) INSTALLED
                                    201,202,203,204,205,206,207,208,209,210,212,213,214,215,216
=========================================================================================================================
=                                                                                                                       =
=                  I N S T A L L A T I O N   P A R A M E T E R S   ( M 4 P A R A M S , M A R K L I B P )                =
=                                                                                                                       =
=  SYSTEM DELIMITER:   #          PAGE HEIGHT:       66         M4LIST WIDTH:      132        DEF WIDTH OF PAGE:  0    =
=                                                                                                                       =
=  AUTO GRAND:         N          HEADING CHAR:      -          SUBTITLE REPEAT:   N          INVALID FIELD:      *    =
=                                                                                                                       =
=  MISSING FIELD:      -          NON-EDIT FIELD:    +          PERCENT CHAR:      %          LEFT SEPARATOR:     (    =
=                                                                                                                       =
=  RIGHT SEPARATOR:    )          SINGLE SEPARATOR:  ,          SOURCE SPACING:    1          PRINT MESSAGES:     Y    =
=                                                                                                                       =
=  CONSOLE MESSAGES:   N          M4REPO BLOCKSIZE:  4,096      INPUT I/O BUFFERS: 2          OUTPUT I/O BUFFERS: 1    =
=                                                                                                                       =
=  SNGL-STEP STORAGE:  8,192      SNGL-STEP SORTSIZE: 524,288   DIGIT SELECT CHAR: 9          ZERO SUPPRESS CHAR: Z    =
=                                                                                                                       =
=  CURRENCY CHAR:      $          PLUS CHAR:         +          MINUS CHAR:        -          CHECK PROTECT CHAR: *    =
=                                                                                                                       =
```

Figure D-2 R06PREPA (Page 24 of 26)

```
=  DECIMAL CHAR:       .          GROUPING CHAR:      ,          PRIMARY PLOT CHAR:  X          SECONDARY PLOT CHAR:*          =
=                                                                                                                               =
=  FIT PLOT CHAR:      .          HORIZONTAL AXIS:    _          HORIZONTAL HASH:    |          VERTICAL AXIS:     |          =
=                                                                                                                               =
=  VERTICAL HASH:      -          MINUTES/HOUR:       60         SECONDS/MINUTE:     60         TIME DELIMITER:     HH:MM:SS =
=                                                                                                                               =
=  DATE FORMAT:      MMM DD, YYYY TODAY FORMAT+DELIM: MM/DD/YY   ISDATE DELIMITER:   YYYY-MM-DD JULIAN DELIMITER:   YY.DDD   =
=                                                                                                                               =
=  SORT PROGRAM CODE:  2          MINCORE VALUE:      12 K       ALT M4LIST WIDTH:   264        ALT DEF W/OF PAGE:  0          =
=                                                                                                                               =
=  MAX LINES OF TRACE: 1,024      ITEM TRACKING:      0          SUPPRESS NDS REPT?: N          DEFAULT MAXGETMN:   1,024 K    =
=                                                                                                                               =
=  CONDITION CODE 1:   0          CONDITION CODE 2:   4          CONDITION CODE 3:   0          CONDITION CODE 4:   16         =
=                                                                                                                               =
================================================================================================================================
NOV 03, 2001  13.20.27                                                                                                  PAGE     2
--------------------------------------------------------------------------------------------------------------------------------
   RUN     STMT    FILE    O N S D U T R A B S D S L  RB  R S M R S            START           END             SEQUENCE   LIST
   NAME    TYPE    NAME    L E E I P R E U U R L C S  PL  J O O F S            SEARCH          SEARCH          73 TO 80   SEQ
                           D W Q R D N P D F T M N T  TK  T P P O R
--------------------------------------------------------------------------------------------------------------------------------
(BLDIFILE) (RC) (ITEMDAT1)(S)          (S)        (T,G)          (R)                                          (ASLGEN'D)     1
    CONTROL NAME BLDIFILE TERM LISTGEN
    FILE MASTER INPUT  ITEMDAT1
    FILE REPORT
--------------------------------------------------------------------------------------------------------------------------------
   RUN     STMT  LOGICAL  LBL  DTF/DD      KEY       KEY       KEY      QUALIFIER   USER I/O  ARRAY       PASSWORD/  SEQUENCE   LIST
   NAME    TYPE  FILENAME       NAME      FIELD-1   FIELD-2   FIELD-3   ICF CRD DBI   MODULE   QUAL        AUTH.-ID   73 TO 80   SEQ
--------------------------------------------------------------------------------------------------------------------------------
(BLDIFILE) (RF) (ITEMDAT1)(S)(M4OLD    )                                                                    (ASLGEN'D)     2
```

```
================================================================================================================================
=                                                                                                                              =
=                                              F I L E    S U M M A R Y                                                        =
=                                                                                                                              =
= JOBNAME: KREJO04X                          STEPNAME: PROC4                                                                   =
=                                                                                                                              =
=  FILE NAME: M4OLD                          DSNAME: KREJO04.KREJO04X.JOB04981.D0000109.?                                      =
=  FILE NAME: M4LIB                          DSNAME: BUILDER.R140.M4LIB.R06#1                                                  =
=                                                                                                                              =
================================================================================================================================
NOV 03, 2001  13.20.27                                                                                                  PAGE     3
                                              ****************************
                                              *  PROC NAME - REPT__1     *
                                              *  INPUT STREAM PROCEDURE  *
                                              ****************************
--------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                              MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID        ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM       73 TO 80   SEQ
--------------------------------------------------------------------------------------------------------------------------------
(ER)                                                                          (N)                              (ASLGEN'D)     3
    REPORT ITEMNO ITEMNAME ITMPRICE ITMWEIGH
      TITLE 'BUILD THE ITEM FILE - A VSAM DATA SET'
      TITLE 'LIST THE DATA OUTPUT TO THE VSAM ITEM FILE'
    END REPORT
--------------------------------------------------------------------------------------------------------------------------------
 ST SUM V  8  PG  PG   I SP  MAXIMUM TB COLUMN  DT  PG  STRT LIN SL RH .......SUBFILE..... C E A S R S N      C  SEQUENCE   LIST
 TY RPT S LPI WID HGT  M FRM LPP PGS TO HEADING POS POS PAGE NUM UB PN ENT    NAME   BLK/ FMT H F G T S U D      O  73 TO 80   SEQ
 MP     P             G            LT TYP POS         NUM      ML TD REC           BUF      C C S L T M S      N
--------------------------------------------------------------------------------------------------------------------------------
(E1)                                                                                                           (ASLGEN'D)     4
--------------------------------------------------------------------------------------------------------------------------------
        SP  Q              E  N  S  D  C  S  T  C  C  M  M  A  G  D Q PCT/RAT     P              PARTIAL   SEQUENCE   LIST
STMT SEQ  BF  L  FIELD  SEG-  N  P  E  E  T  U  O  U  N  A  I  V  M  E L  FIELD  RA- C     PICTURE      STRT NO   73 TO 80   SEQ
TYPE NO.  COL F  NAME     LVL  D  R  Q  S  L  B  T  M  T  X  N  G  R  C F  NAME   TIO T      EDIT        CHR CHR
--------------------------------------------------------------------------------------------------------------------------------
(R1)          (ITEMNO  ) 001-1                                                                               (ASLGEN'D)     5
(R1)          (ITEMNAME) 001-1                                                                               (ASLGEN'D)     6
(R1)          (ITMPRICE) 001-1                                                                               (ASLGEN'D)     7
(R1)          (ITMWEIGH) 001-1                                                                               (ASLGEN'D)     8
--------------------------------------------------------------------------------------------------------------------------------
STMT  SEQ                                                                                                      SEQUENCE   LIST
TYPE  NO.                                        TEXT                                                          73 TO 80   SEQ
--------------------------------------------------------------------------------------------------------------------------------
(T1) (101)                 (BUILD THE ITEM FILE - A VSAM DATA SET#                  )                          (ASLGEN'D)     9
(T1) (102)                 (LIST THE DATA OUTPUT TO THE VSAM ITEM FILE#             )                          (ASLGEN'D)    10
NOV 03, 2001  13.20.27                                                                                                  PAGE     4
                                              ****************************
                                              *  PROC NAME - SUBF__2     *
                                              *  INPUT STREAM PROCEDURE  *
                                              ****************************
--------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                              MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID        ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM       73 TO 80   SEQ
```

Figure D-2 R06PREPA (Page 25 of 26)

```
-------------------------------------------------------------------------------------------------------------------------------
(ER)                                                                (N)                                  (ASLGEN'D)    11
     EXTRACT DDNAME ITMFILE RECFM KEYVSAM,
             ITEMS ITEMNO ITEMNAME ITMPRICE ITMWEIGH
-------------------------------------------------------------------------------------------------------------------------------
 ST SUM V  8  PG  PG   I SP  MAXIMUM TB COLUMN  DT  PG  STRT LIN SL RH .......SUBFILE........ C E A S R S N      C  SEQUENCE   LIST
 TY RPT S LPI WID HGT  M FRM LPP PGS TO HEADING POS POS PAGE NUM UB PN ENT    NAME   BLK/ FMT H F G T S U D      O  73 TO 80    SEQ
 MP     P            G           LT TYP POS         NUM      ML TD REC           BUF      C C S L T M S      N
-------------------------------------------------------------------------------------------------------------------------------
(E1)                                                          (DO)   (ITMFILE )      (K)                     (ASLGEN'D)    12
-------------------------------------------------------------------------------------------------------------------------------
         SP  Q              E N S  D C S T  C  C M M A  G  D Q PCT/RAT     P              PARTIAL   SEQUENCE   LIST
STMT SEQ BF  L  FIELD  SEG- N P E  E T U O  U  N A I V  M  R E L FIELD    RA- C     PICTURE    STRT NO  73 TO 80    SEQ
TYPE NO. COL F  NAME     LVL D R Q  S L B T  M  T X N G  R  C F NAME     TIO T      EDIT      CHR CHR
-------------------------------------------------------------------------------------------------------------------------------
(R1)          (ITEMNO  ) 001-1                                                                 (ASLGEN'D)    13
(R1)          (ITEMNAME) 001-1                                                                 (ASLGEN'D)    14
(R1)          (ITMPRICE) 001-1                                                                 (ASLGEN'D)    15
(R1)          (ITMWEIGH) 001-1                                                                 (ASLGEN'D)    16
** MK4TY05  TYPE 0  SUBFILE RECORD LENGTH COMPUTED FROM THE ABOVE SPECIFICATION IS    38.                                *****
NOV 03, 2001  13.20.27                                                                                PAGE      6
** MK4W204  TYPE 0  NUMBER OF MESSAGES PRINTED IS    2.                                                          *****
                   M4OLD    -       17 RECORDS INPUT
                   M4REPO   -       30 RECORDS OUTPUT
                   M4INPUT  -        9 RECORDS INPUT
                   M4LIST   -      140 RECORDS OUTPUT
                   ITMFILE  -       17 RECORDS OUTPUT
                     15 TRACKS ASSIGNED TO M4LIB    --   13 TRACKS NOT FULL. LIBRARY DIRECTORY BLOCKING FACTOR IS  800.
** MK4W209  TYPE 0  1016872 BYTES OF MAIN STORAGE UNUSED DURING DECODING PHASE                                      *****
                   1017424 BYTES OF MAIN STORAGE UNUSED DURING COMPILATION PHASE
                    564896 BYTES OF MAIN STORAGE UNUSED DURING PROCESSING PHASE
** MK4ED02  TYPE 0  REQUEST-REPT___1  REPORT-1  REQUESTOR ID-                                                      *****
                                        BUILD THE ITEM FILE - A VSAM DATA SET                              PAGE 1
                                      LIST THE DATA OUTPUT TO THE VSAM ITEM FILE
-------------------------------------------------------------------------------------------------------------------------------
   ITEM           ITEM           ITEM      ITEM
  NUMBER          NAME           PRICE     WEIGH
-------------------------------------------------------------------------------------------------------------------------------
  BL-B01B  BLDR QUICK REF           $4.50      .75
  BL-H11A  BLDR HANDBOOK ASL        31.00     3.25
  BL-H11B  BLDR HANDBOOK FIXED      35.00     3.75
  BL-T01B  BLDR TRAINING BASIC     475.00    17.75
  BL-T02B  BLDR TRAINING ADVAN     535.00    23.75
  E13      INFORM CICS GUIDE        28.95     3.25
  E68      BUILDER CONCEPTS         27.50     5.25
  H11      BLDR HANDBOOK            25.50     2.25
  H68      BLDR QUICK REF            7.95      .35
  M52      TRANSACT CONCEPTS        19.95     4.50
  M55      TRANSACT INSTALL         15.95     1.00
  N11      INFORM CONCEPTS          23.50     5.15
  N19      INFORM INSTALL           18.95     1.30
  N34      DATA BASE CONCEPTS       39.95     3.25
  N94      TRANSACT CICS GUIDE      39.95     3.25
  O13      TRANSACT QUICK REF        7.95      .25
  O17      TRANSACT INST UPDS        3.95      .40
** MK4FT03  TYPE 0  END OF REPORT.                                                                                 *****
** MK4W204  TYPE 0  NUMBER OF MESSAGES PRINTED IS    2.                                                          *****
                   M4REPO   -       29 RECORDS INPUT
                   M4LIST   -       27 RECORDS OUTPUT
** MK4W209  TYPE 0  1005396 BYTES OF MAIN STORAGE UNUSED DURING REPORTING PHASE                                    *****

DATA SET UTILITY - GENERATE                                                                  PAGE 0001
   GENERATE MAXFLDS=10
     RECORD FIELD=(50)

PROCESSING ENDED AT EOD
DATA SET UTILITY - GENERATE                                                                  PAGE 0001
   GENERATE MAXFLDS=10
     RECORD FIELD=(50)

PROCESSING ENDED AT EOD
IDCAMS  SYSTEM SERVICES                                    TIME: 13:20:29       11/03/01     PAGE      1

  REPRO INFILE(SEQL) -
    OUTDATASET(BUILDER.R140.CORD.FILE.R06#1)  /* <<< D... S.. N...   */
IDC0005I NUMBER OF RECORDS PROCESSED WAS 9
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0

                                          /* <<< S.. S... IDCAMS */
IDC0002I IDCAMS PROCESSING COMPLETE. MAXIMUM CONDITION CODE WAS 0
```

Figure D-2 R06PREPA (Page 26 of 26)

## Examples Jobs

The following item is some of the Output Listing from the Examples job stream in Chapter 6, *Input File Processing*. The main reference label and associated member name is R06FIGSA.

The Examples job stream contains several VISION:Builder applications that demonstrate the coding and produce the outputs described and shown in Chapter 6, *Input File Processing*. The actual JCL, source code, and data is delivered on the VISION:Builder system tape in the Samples file. This file is downloaded during the installation process. Refer to the *VISION:Builder Installation Guide* for more information.

### Indexed Coordination to Produce Shipping Lists: The Banner Page

```
NOV 03, 2001  13.23.46                                                                                    PAGE      1
                                          ********************************************
                                          *  VISION:Builder   4400  (OS/390 - 14.0)  *
                                          *            COPYRIGHT (C) 2002            *
                                          *  COMPUTER ASSOCIATES INTERNATIONAL, INC. *
                                          ********************************************
 VISION:Builder JCR1 14.0 110201              ENABLED FOR IBM LANGUAGE ENVIRONMENT                BUILD STAMP = 101306,12:51:24.
                                         COMLIB  MODIFICATIONS (PTFs, APARs) INSTALLED
                                   201,202,203,204,205,206,207,208,209,210,212,213,214,215,216
 ==================================================================================================================================
 =                                                                                                                                =
 =                  I N S T A L L A T I O N   P A R A M E T E R S   ( M 4 P A R A M S , M A R K L I B P )                        =
 =                                                                                                                                =
 =  SYSTEM DELIMITER:   #          PAGE HEIGHT:        66         M4LIST WIDTH:       132        DEF WIDTH OF PAGE:  0          =
 =                                                                                                                                =
 =  AUTO GRAND:         N          HEADING CHAR:       -          SUBTITLE REPEAT:    N          INVALID FIELD:      *          =
 =                                                                                                                                =
 =  MISSING FIELD:      -          NON-EDIT FIELD:     +          PERCENT CHAR:       %          LEFT SEPARATOR:     (          =
 =                                                                                                                                =
 =  RIGHT SEPARATOR:    )          SINGLE SEPARATOR:   ,          SOURCE SPACING:     1          PRINT MESSAGES:     Y          =
 =                                                                                                                                =
 =  CONSOLE MESSAGES:   N          M4REPO BLOCKSIZE:   4,096      INPUT I/O BUFFERS:  2          OUTPUT I/O BUFFERS: 1          =
 =                                                                                                                                =
 =  SNGL-STEP STORAGE:  8,192      SNGL-STEP SORTSIZE: 524,288    DIGIT SELECT CHAR:  9          ZERO SUPPRESS CHAR: Z          =
 =                                                                                                                                =
 =  CURRENCY CHAR:      $          PLUS CHAR:          +          MINUS CHAR:         -          CHECK PROTECT CHAR: *          =
 =                                                                                                                                =
 =  DECIMAL CHAR:       .          GROUPING CHAR:      ,          PRIMARY PLOT CHAR:  X          SECONDARY PLOT CHAR:*          =
 =                                                                                                                                =
 =  FIT PLOT CHAR:      .          HORIZONTAL AXIS:    _          HORIZONTAL HASH:    |          VERTICAL AXIS:      |          =
 =                                                                                                                                =
 =  VERTICAL HASH:      -          MINUTES/HOUR:       60         SECONDS/MINUTE:     60         TIME DELIMITER:     HH:MM:SS   =
 =                                                                                                                                =
 =  DATE FORMAT:      MMM DD, YYYY  TODAY FORMAT+DELIM: MM/DD/YY   ISDATE DELIMITER:   YYYY-MM-DD  JULIAN DELIMITER:   YY.DDD     =
 =                                                                                                                                =
 =  SORT PROGRAM CODE:  2          MINCORE VALUE:      12 K       ALT M4LIST WIDTH:   264        ALT DEF W/OF PAGE:  0          =
 =                                                                                                                                =
 =  MAX LINES OF TRACE: 1,024      ITEM TRACKING:      0          SUPPRESS NDS REPT?: N          DEFAULT MAXGETMN:   1,024 K    =
 =                                                                                                                                =
 =  CONDITION CODE 1:   0          CONDITION CODE 2:   4          CONDITION CODE 3:   0          CONDITION CODE 4:   16         =
 =                                                                                                                                =
 ==================================================================================================================================
```

Figure D-3 Indexed Coordination to Produce Shipping Lists: The Banner Page

## Indexed Coordination to Produce Shipping Lists: The Run Control Commands – Part 1 of 5

```
NOV 03, 2001  13.23.46                                                                                                  PAGE      2
-------------------------------------------------------------------------------------------------------------------------------------
   RUN     STMT    FILE    O N S D U T R A B S D S L  RB  R S M R S              START            END              SEQUENCE   LIST
   NAME    TYPE    NAME    L E E I P R E U U R L C S  PL  J O O F S              SEARCH           SEARCH           73 TO 80    SEQ
                           D W Q R D N P D F T M N T  TK  T P P O R
-------------------------------------------------------------------------------------------------------------------------------------
(F144    ) (RC)  (CUSTOMER) (S)          (S)        (T,G)          (R)                                                (ASLGEN'D)     1
     CONTROL NAME F144 TERM LISTGEN
     FILE MASTER INPUT  CUSTOMER
     FILE CORD1 ITEMFILE DIRECT BY O.ITEMORD
     FILE REPORT
-------------------------------------------------------------------------------------------------------------------------------------
   RUN     STMT  LOGICAL  LBL  DTF/DD      KEY       KEY       KEY      QUALIFIER   USER I/O  ARRAY       PASSWORD/  SEQUENCE   LIST
   NAME    TYPE  FILENAME       NAME     FIELD-1   FIELD-2   FIELD-3   ICF CRD DBI   MODULE   QUAL        AUTH.-ID   73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------------
(F144    ) (RF)  (CUSTOMER) (S) (M4OLD   )                                                                          (ASLGEN'D)     2
(F144    ) (RF)  (ITEMFILE) (S) (M4CORD1 ) (ITEMORD )                    (O)                                       (ASLGEN'D)     3


=====================================================================================================================================
=                                                                                                                                   =
=                                                   F I L E   S U M M A R Y                                                         =
=                                                                                                                                   =
= JOBNAME: KREJO04X                                STEPNAME: F144                                                                   =
=                                                                                                                                   =
=  FILE NAME: M4OLD                                DSNAME: BUILDER.R140.CUSTOMER.MASTER.R06#1                                       =
=  FILE NAME: M4CORD1                              DSNAME: BUILDER.R140.ITEM.FILE.R06#1                                             =
=  FILE NAME: M4LIB                                DSNAME: BUILDER.R140.M4LIB.R06#1                                                 =
=                                                                                                                                   =
=====================================================================================================================================
```

Figure D-4 Indexed Coordination to Produce Shipping Lists: The Run Control Commands Part 1

## Indexed Coordination to Produce Shipping Lists: The PROCEDURE Commands – Part 2 of 5

```
NOV 03, 2001  13.23.46                                                                                                  PAGE      3
                                                  ******************************
                                                  *  PROC NAME - MAIN          *
                                                  *  INPUT STREAM PROCEDURE    *
                                                  ******************************
-------------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                                  MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE           REQUESTOR ID           ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM       73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------------
(ER)                                                                                  (N)                              (ASLGEN'D)     4
     MAIN: PROC
     ;
     SHPWEIGH: FIELD P 5 2 HEADING 'SHIPPING' 'WEIGHT'
     ;
     DO FORALL O.ITEMORD
     ;
      CALL REPORT RPTITEMS
     ;
      LET T.SHPWEIGH = O.ITMQTYOR * 1.ITMWEIGH
      CALL REPORT RPTWEIGH
     ;
     END DO
     ;
     END PROC
     ;
-------------------------------------------------------------------------------------------------------------------------------------
STMT   FIELD    FIELD FLD DEC OUTPT EDIT      INITIAL          LINE 1 OF        LINE 2 OF                            SEQUENCE   LIST
TYPE   NAME     LNGTH TYP PLC EDIT  LGTH       VALUE         COLUMN HEADING   COLUMN HEADING                         73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------------
(TF) (SHPWEIGH) (  5) (P) (2)                                (SHIPPING      ) (WEIGHT        )                        (ASLGEN'D)     5
-------------------------------------------------------------------------------------------------------------------------------------
STMT SEQ  LOG CON .....OPERAND-A.....  OPER  ............OPERAND-B.............   ......RESULT......  PARTIAL FIELD  SEQUENCE   LIST
TYPE NO.  LEV CTR QLF FIELD   SEG-LVL  ATN   QLF FIELD   SEG-LVL  (OR CONSTANT)   QLF FIELD  SEG-LVL  STRT NUMB OPR  73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------------
(PR) (100)                             (GO)     (SUB LOOP___2                  )                                      (ASLGEN'D)     6
NOV 03, 2001  13.23.46                                                                                                  PAGE      4
                                                  ******************************
                                                  *  REQUEST NAME - LOOP___2   *
                                                  *  GENERATED REQUEST         *
```

Figure D-5 Indexed Coordination to Produce Shipping Lists: The PROCEDURE Commands Part 2 (Page 1 of 2)

```
                                    ****************************
-------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                              MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID        ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM      73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(ER)                                                                          (S)                              (ASLGEN'D)     7
-------------------------------------------------------------------------------------------------------------------------------
STMT SEQ  LOG CON .....OPERAND-A.....  OPER  .............OPERAND-B.............  ......RESULT......  PARTIAL FIELD  SEQUENCE   LIST
TYPE NO.  LEV CTR QLF FIELD   SEG-LVL  ATN   QLF FIELD   SEG-LVL  (OR CONSTANT)   QLF FIELD  SEG-LVL  STRT NUMB OPR  73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(PR) (100)         (O,ITEMORD ) 005-3  (EQ)    (O,ITEMORD ) 005-3                                                 (ASLGEN'D)     8

(PR) (101)                             (NS)      (RETURN                    )                                    (ASLGEN'D)     9
(PR) (102)                             (GO)      (SUB RPTITEMS              )                                    (ASLGEN'D)    10
(PR) (103)         (O,ITMQTYOR) 005-3  (* )    (1,ITMWEIGH) 001-1             (T,SHPWEIGH)                       (ASLGEN'D)    11
(PR) (104)                             (GO)      (SUB RPTWEIGH              )                                    (ASLGEN'D)    12
```

Figure D-5 Indexed Coordination to Produce Shipping Lists: The PROCEDURE Commands Part 2 (Page 1 of 2)

## Indexed Coordination to Produce Shipping Lists: Report 1 REPORT Commands – Part 3 of 5

```
NOV 03, 2001  13.23.46                                                                                          PAGE     5
                                                    ****************************
                                                    *  PROC NAME - RPTITEMS    *
                                                    *  INPUT STREAM PROCEDURE  *
                                                    ****************************
-------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                              MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID        ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM      73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(ER) (TODAY )                                                                 (S)                              (ASLGEN'D)    13
     RPTITEMS: REPORT O.CUSTNO O.ORDERNO O.ITEMORD,
                       1.ITEMNAME O.ITMQTYOR,
       TYPE SUBROUTINE
       ORDER BY O.CUSTNO
       GROUP BY O.CUSTNO
       TOTAL O.ITMQTYOR BY O.CUSTNO
       TITLE 'SHIPPING LIST REPORT - ITEMS ORDERED PER CUSTOMER'
       TITLE 'DIRECT READ ITEM FILE FOR NAME'
       FORMAT SUMMARYLABELS SUPPRESS
       END REPORT
     ;
-------------------------------------------------------------------------------------------------------------------------------
 ST SUM V  8  PG  PG   I SP  MAXIMUM TB COLUMN  DT  PG  STRT LIN SL RH .......SUBFILE........ C E A S R S N      C  SEQUENCE   LIST
 TY RPT S LPI WID HGT  M FRM LPP PGS TO HEADING POS POS PAGE NUM UB PN ENT    NAME   BLK/ FMT H F G T S U D      O  73 TO 80    SEQ
 MP     P             G           LT TYP POS        NUM      ML TD REC          BUF      C C S L T M S      N
-------------------------------------------------------------------------------------------------------------------------------
(E1)                                                          (X)                                            (ASLGEN'D)    14
-------------------------------------------------------------------------------------------------------------------------------
        SP  Q              E N S D C S T C C M M A G D Q PCT/RAT     P                PARTIAL   SEQUENCE   LIST
STMT SEQ BF  L  FIELD   SEG- N P E E T U O U N A I V M E L  FIELD   RA- C     PICTURE    STRT NO  73 TO 80    SEQ
TYPE NO. COL F  NAME     LVL D R Q S L B T M T X N G R C F  NAME    TIO T      EDIT       CHR CHR
-------------------------------------------------------------------------------------------------------------------------------
(R1)        (O,CUSTNO  ) 001-1      (1)   (1)                                                                 (ASLGEN'D)    15
(R1)        (O,ORDERNO ) 002-2                                                                                (ASLGEN'D)    16
(R1)        (O,ITEMORD ) 005-3                                                                                (ASLGEN'D)    17
(R1)        (1,ITEMNAME) 001-1                                                                                (ASLGEN'D)    18
(R1)        (O,ITMQTYOR) 005-3                (1)                                                             (ASLGEN'D)    19
-------------------------------------------------------------------------------------------------------------------------------
STMT  SEQ                                                                                                       SEQUENCE   LIST
TYPE  NO.                                          TEXT                                                         73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(T1)  (101)              (SHIPPING LIST REPORT - ITEMS ORDERED PER CUSTOMER#        )                          (ASLGEN'D)    20
(T1)  (102)              (DIRECT READ ITEM FILE FOR NAME#                           )                          (ASLGEN'D)    21
NOV 03, 2001  13.23.46                                                                                          PAGE     6
                                                    ****************************
                                                    *  PROC NAME - RPTWEIGH    *
                                                    *  INPUT STREAM PROCEDURE  *
                                                    ****************************
-------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                              MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID        ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM      73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(ER) (TODAY )                                                                 (S)                              (ASLGEN'D)    22
```

Figure D-6 Indexed Coordination to Produce Shipping Lists: Report 1 REPORT Commands Part 3

## Indexed Coordination to Produce Shipping Lists: Report 2 REPORT Commands – Part 4 of 5

```
     RPTWEIGH: REPORT O.CUSTNO O.ORDERNO O.ITEMORD,
                      1.ITEMNAME 1.ITMWEIGH O.ITMQTYOR T.SHPWEIGH,
       TYPE SUBROUTINE
       ORDER BY O.CUSTNO
       GROUP BY O.CUSTNO
       TOTAL O.ITMQTYOR BY O.CUSTNO
       TOTAL T.SHPWEIGH BY O.CUSTNO
       TITLE 'SHIPPING LIST REPORT - TOTAL WEIGHT PER CUSTOMER'
       TITLE 'DIRECT READ ITEM FILE FOR NAME AND WEIGHT'
       FORMAT SUMMARYLABELS SUPPRESS
       END REPORT
-------------------------------------------------------------------------------------------------------------------------------
 ST SUM V  8  PG  PG   I SP  MAXIMUM TB COLUMN  DT  PG  STRT LIN SL RH .......SUBFILE........ C E A S R S N      C  SEQUENCE   LIST
 TY RPT S LPI WID HGT  M FRM LPP PGS TO HEADING POS POS PAGE NUM UB PN ENT    NAME   BLK/ FMT H F G T S U D      O  73 TO 80    SEQ
 MP     P             G            LT TYP POS         NUM      ML TD REC          BUF      C C S L T M S      N
-------------------------------------------------------------------------------------------------------------------------------
(E1)                                                          (X)                                              (ASLGEN'D)    23
-------------------------------------------------------------------------------------------------------------------------------
         SP  Q              E N  S  D  C  S  T  C  C  M  M  A  G  D Q PCT/RAT      P               PARTIAL   SEQUENCE   LIST
STMT SEQ BF  L  FIELD   SEG- N P  E  E  T  U  O  U  N  A  I  V  M  R E L FIELD    RA- C     PICTURE     STRT NO   73 TO 80   SEQ
TYPE NO. COL F  NAME      LVL D R  Q  S  L  B  T  M  T  X  N  G  R  C F NAME     TIO T      EDIT       CHR CHR
-------------------------------------------------------------------------------------------------------------------------------
(R1)        (O,CUSTNO  ) 001-1      (1)   (1)                                                          (ASLGEN'D)    24
(R1)        (O,ORDERNO ) 002-2                                                                         (ASLGEN'D)    25
(R1)        (O,ITEMORD ) 005-3                                                                         (ASLGEN'D)    26
(R1)        (1,ITEMNAME) 001-1                                                                         (ASLGEN'D)    27
(R1)        (1,ITMWEIGH) 001-1                                                                         (ASLGEN'D)    28
(R1)        (O,ITMQTYOR) 005-3                  (1)                                                    (ASLGEN'D)    29
(R1)        (T,SHPWEIGH)                        (1)                                                    (ASLGEN'D)    30
-------------------------------------------------------------------------------------------------------------------------------
STMT  SEQ                                                                                                SEQUENCE   LIST
TYPE  NO.                                          TEXT                                                  73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(T1) (101)              (SHIPPING LIST REPORT - TOTAL WEIGHT PER CUSTOMER#         )                      (ASLGEN'D)    31
(T1) (102)              (DIRECT READ ITEM FILE FOR NAME AND WEIGHT#                )                      (ASLGEN'D)    32
```

Figure D-7 Indexed Coordination to Produce Shipping Lists: Report 2 REPORT Commands Part 4

## Indexed Coordination to Produce Shipping Lists: Report 2 REPORT Commands – Part 5 of 5

```
NOV 03, 2001  13.23.46                                                                                          PAGE      7
** MK4W204  TYPE 0  NUMBER OF MESSAGES PRINTED IS    0.                                                                *****
                    M4OLD    -       5 RECORDS INPUT
                    M4CORD1  -       5 RECORDS INPUT
                    M4REPO   -      44 RECORDS OUTPUT
                    M4INPUT  -      41 RECORDS INPUT
                    M4LIST   -     221 RECORDS OUTPUT
                      15 TRACKS ASSIGNED TO M4LIB    --   13 TRACKS NOT FULL. LIBRARY DIRECTORY BLOCKING FACTOR IS  800.
** MK4W209  TYPE 0  1013448 BYTES OF MAIN STORAGE UNUSED DURING DECODING PHASE                                       *****
                    1014816 BYTES OF MAIN STORAGE UNUSED DURING COMPILATION PHASE
                     565008 BYTES OF MAIN STORAGE UNUSED DURING PROCESSING PHASE
** MK4ED02  TYPE 0  REQUEST-RPTITEMS  REPORT-1  REQUESTOR ID-                                                       *****
11/03/01                                SHIPPING LIST REPORT - ITEMS ORDERED PER CUSTOMER                       PAGE 1
                                           DIRECT READ ITEM FILE FOR NAME
-------------------------------------------------------------------------------------------------------------------------------
  CUSTOMER  ORDER   STOCK NO.          ITEM          ORDER
    NUMBER  NUMBER   ORDERED          NAME         QUANTITY
-------------------------------------------------------------------------------------------------------------------------------
    00002   20103    BL-H11A   BLDR HANDBOOK ASL            3
            20105    BL-H11B   BLDR HANDBOOK FIXED          3
    00002                                                   6
    00003   30101    BL-B01B   BLDR QUICK REF               5
    00003                                                   5
    00010   01101    BL-T01B   BLDR TRAINING BASIC          1
            01104    BL-T02B   BLDR TRAINING ADVAN          1
    00010                                                   2
** MK4FT03  TYPE 0  END OF REPORT.                                                                                  *****
** MK4ED02  TYPE 0  REQUEST-RPTWEIGH  REPORT-1  REQUESTOR ID-                                                       *****
11/03/01                                SHIPPING LIST REPORT - TOTAL WEIGHT PER CUSTOMER                        PAGE 1
                                           DIRECT READ ITEM FILE FOR NAME AND WEIGHT
-------------------------------------------------------------------------------------------------------------------------------
  CUSTOMER  ORDER   STOCK NO.          ITEM          ITEM     ORDER      SHIPPING
    NUMBER  NUMBER   ORDERED          NAME         WEIGH   QUANTITY     WEIGHT
-------------------------------------------------------------------------------------------------------------------------------
    00002   20103    BL-H11A   BLDR HANDBOOK ASL      3.25          3          9.75
```

Figure D-8 Indexed Coordination to Produce Shipping Lists: Report 2 REPORT Commands Part 5 (Page 1 of 2)

```
          20105    BL-H11B   BLDR HANDBOOK FIXED      3.75          3          11.25
   00002                                                            6          21.00
   00003   30101    BL-B01B   BLDR QUICK REF            .75          5           3.75
   00003                                                            5           3.75
   00010   01101    BL-T01B   BLDR TRAINING BASIC     17.75          1          17.75
           01104    BL-T02B   BLDR TRAINING ADVAN     23.75          1          23.75
   00010                                                            2          41.50
** MK4FT03  TYPE 0  END OF REPORT.                                                                                  *****
** MK4W204  TYPE 0  NUMBER OF MESSAGES PRINTED IS    4.                                                             *****
                    M4REPO   -       43 RECORDS INPUT
                    M4LIST   -       34 RECORDS OUTPUT
** MK4W209  TYPE 0  1004560 BYTES OF MAIN STORAGE UNUSED DURING REPORTING PHASE                                     *****
```

Figure D-8 Indexed Coordination to Produce Shipping Lists: Report 2 REPORT Commands Part 5 (Page 2 of 2)

## Single-File Sequential Coordination Match Only – Banner Page

```
NOV 03, 2001  13.23.47                                                                                    PAGE      1
                                          ********************************************
                                          *  VISION:Builder   4400  (OS/390 - 14.0)  *
                                          *            COPYRIGHT (C) 2002            *
                                          *  COMPUTER ASSOCIATES INTERNATIONAL, INC. *
                                          ********************************************
VISION:Builder JCR1 14.0 110201                ENABLED FOR IBM LANGUAGE ENVIRONMENT                BUILD STAMP = 101306,12:51:24.
                                          COMLIB  MODIFICATIONS (PTFs, APARs) INSTALLED
                                   201,202,203,204,205,206,207,208,209,210,212,213,214,215,216
==================================================================================================================================
=                                                                                                                                =
=                  I N S T A L L A T I O N   P A R A M E T E R S   ( M 4 P A R A M S , M A R K L I B P )                        =
=                                                                                                                                =
=  SYSTEM DELIMITER:   #          PAGE HEIGHT:        66         M4LIST WIDTH:       132        DEF WIDTH OF PAGE:  0          =
=                                                                                                                                =
=  AUTO GRAND:         N          HEADING CHAR:       -          SUBTITLE REPEAT:    N          INVALID FIELD:      *          =
=                                                                                                                                =
=  MISSING FIELD:      -          NON-EDIT FIELD:     +          PERCENT CHAR:       %          LEFT SEPARATOR:     (          =
=                                                                                                                                =
=  RIGHT SEPARATOR:    )          SINGLE SEPARATOR:   ,          SOURCE SPACING:     1          PRINT MESSAGES:     Y          =
=                                                                                                                                =
=  CONSOLE MESSAGES:   N          M4REPO BLOCKSIZE:   4,096      INPUT I/O BUFFERS:  2          OUTPUT I/O BUFFERS: 1          =
=                                                                                                                                =
=  SNGL-STEP STORAGE:  8,192      SNGL-STEP SORTSIZE: 524,288    DIGIT SELECT CHAR:  9          ZERO SUPPRESS CHAR: Z          =
=                                                                                                                                =
=  CURRENCY CHAR:      $          PLUS CHAR:          +          MINUS CHAR:         -          CHECK PROTECT CHAR: *          =
=                                                                                                                                =
=  DECIMAL CHAR:       .          GROUPING CHAR:      ,          PRIMARY PLOT CHAR:  X          SECONDARY PLOT CHAR:*          =
=                                                                                                                                =
=  FIT PLOT CHAR:      .          HORIZONTAL AXIS:    _          HORIZONTAL HASH:    |          VERTICAL AXIS:      |          =
=                                                                                                                                =
=  VERTICAL HASH:      -          MINUTES/HOUR:       60         SECONDS/MINUTE:     60         TIME DELIMITER:     HH:MM:SS   =
=                                                                                                                                =
=  DATE FORMAT:      MMM DD, YYYY TODAY FORMAT+DELIM: MM/DD/YY   ISDATE DELIMITER:   YYYY-MM-DD  JULIAN DELIMITER:   YY.DDD     =
=                                                                                                                                =
=  SORT PROGRAM CODE:  2          MINCORE VALUE:      12 K       ALT M4LIST WIDTH:   264        ALT DEF W/OF PAGE:  0          =
=                                                                                                                                =
=  MAX LINES OF TRACE: 1,024      ITEM TRACKING:      0          SUPPRESS NDS REPT?: N          DEFAULT MAXGETMN:   1,024 K    =
=                                                                                                                                =
=  CONDITION CODE 1:   0          CONDITION CODE 2:   4          CONDITION CODE 3:   0          CONDITION CODE 4:   16         =
=                                                                                                                                =
==================================================================================================================================
```

Figure D-9 Single-File Sequential Coordination Match Only – Banner Page

## Single-File Sequential Coordination Match Only: The Run Control Commands – Part 1 of 2

```
NOV 03, 2001  13.23.47                                                                                    PAGE      2
----------------------------------------------------------------------------------------------------------------------------------
   RUN     STMT    FILE    O N S D U T R A B S D S L  RB  R S M R S            START           END             SEQUENCE   LIST
   NAME    TYPE    NAME    L E E I P R E U U R L C S  PL  J O O F S            SEARCH          SEARCH          73 TO 80    SEQ
                           D W Q R D N P D F T M N T  TK  T P P O R
----------------------------------------------------------------------------------------------------------------------------------
(F150     ) (RC)  (CUSTOMER) (S)          (S)        (T,G)          (R)                                          (ASLGEN'D)     1
     CONTROL NAME F150 TERM LISTGEN
     FILE MASTER INPUT  CUSTOMER
```

Figure D-10 Single-File Sequential Coordination Match Only: The Run Control Commands Part 1 (Page 1 of 2)

```
    FILE CORD1 CUSTHIST MATCHONLY
    FILE REPORT
-------------------------------------------------------------------------------------------------------------------------------------------
  RUN     STMT  LOGICAL  LBL  DTF/DD      KEY       KEY       KEY      QUALIFIER   USER I/O  ARRAY       PASSWORD/  SEQUENCE   LIST
  NAME    TYPE  FILENAME       NAME     FIELD-1   FIELD-2   FIELD-3   ICF CRD DBI   MODULE   QUAL       AUTH.-ID   73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------------------
(F150    ) (RF) (CUSTOMER) (S) (M4OLD    )                                                                          (ASLGEN'D)     2
(F150    ) (RF) (CUSTHIST) (S) (M4CORD1 )                                                                          (ASLGEN'D)     3
```

Figure D-10 Single-File Sequential Coordination Match Only: The Run Control Commands Part 1 (Page 2 of 2)

## Single-File Sequential Coordination Match Only: The Application Coding and Report – Part 2 of 2

```
=============================================================================================================================================
=                                                                                                                                           =
=                                                    F I L E   S U M M A R Y                                                                =
=                                                                                                                                           =
= JOBNAME: KREJO04X                          STEPNAME: F150                                                                                 =
=                                                                                                                                           =
=  FILE NAME: M4OLD                          DSNAME: BUILDER.R140.CUSTOMER.MASTER.R06#1                                                     =
=  FILE NAME: M4CORD1                        DSNAME: BUILDER.R140.CUSTOMER.HISTORY.R06#1                                                    =
=  FILE NAME: M4LIB                          DSNAME: BUILDER.R140.M4LIB.R06#1                                                               =
=                                                                                                                                           =
=============================================================================================================================================
NOV 03, 2001  13.23.47                                                                                                          PAGE      3

                                                  ******************************
                                                  *  PROC NAME - MAIN          *
                                                  *  INPUT STREAM PROCEDURE    *
                                                  ******************************

-------------------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                              MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID        ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM       73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------------------
(ER)                                                                                 (N)                            (ASLGEN'D)     4
    MAIN: PROC
    ;
    COUNT:   FIELD Z 5  HEADING 'CYCLE' 'COUNT'
    COMMENT: FIELD C 25 HEADING 'PROCESS CYCLE' 'IN CONTROL'
    LET F.ECORD = F.ECORD          ; Make a coordinated procedure
    LET T.COUNT = T.COUNT + 1      ; Count the passes thru the procedure
    IF F.MISSPASS EQ ' ' THEN
      LET T.COMMENT = 'SEQUENTIAL COORDINATION'
    ELSE
      LET T.COMMENT = 'STANDARD PROCESSING'
    END IF
    ;
    CALL REPORT DATALIST           ; Go Report the Data
    ;
    END PROC
    ;
-------------------------------------------------------------------------------------------------------------------------------------------
STMT   FIELD    FIELD FLD DEC OUTPT EDIT     INITIAL        LINE 1 OF       LINE 2 OF                                  SEQUENCE   LIST
TYPE   NAME     LNGTH TYP PLC EDIT  LGTH      VALUE        COLUMN HEADING  COLUMN HEADING                             73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------------------
(TF) (COUNT   ) (  5) (Z)                                (CYCLE        ) (COUNT        )                              (ASLGEN'D)     5
(TF) (COMMENT ) ( 25) (C)                                (PROCESS CYCLE ) (IN CONTROL    )                              (ASLGEN'D)     6
-------------------------------------------------------------------------------------------------------------------------------------------
STMT SEQ  LOG CON .....OPERAND-A.....  OPER  .............OPERAND-B.............   ......RESULT......  PARTIAL FIELD  SEQUENCE   LIST
TYPE NO.  LEV CTR QLF FIELD   SEG-LVL  ATN   QLF FIELD   SEG-LVL  (OR CONSTANT)   QLF FIELD  SEG-LVL  STRT NUMB OPR  73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------------------
(PR) (100)                             (R )  (F,ECORD   )                          (F,ECORD   )                    (ASLGEN'D)     7
(PR) (101)          (T,COUNT   )       (+ )  (D,1                               ) (T,COUNT   )                    (ASLGEN'D)     8
(PR) (102)          (F,MISSPASS)       (EQ)  (C)                                                                  (ASLGEN'D)     9
(PR) (103)                             (NS)    (106                             )                                 (ASLGEN'D)    10
(PR) (104)                             (R )  (C,SEQUENTIAL COORDINATION         ) (T,COMMENT )                    (ASLGEN'D)    11
(PR) (105)                             (GO)    (107                             )                                 (ASLGEN'D)    12
(PR) (106)                             (R )  (C,STANDARD PROCESSING             ) (T,COMMENT )                    (ASLGEN'D)    13
(PR) (107)                             (GO)    (SUB DATALIST                    )                                 (ASLGEN'D)    14
NOV 03, 2001  13.23.47                                                                                                          PAGE      4

                                                  ******************************
                                                  *  PROC NAME - DATALIST      *
                                                  *  INPUT STREAM PROCEDURE    *
                                                  ******************************

-------------------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                              MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID        ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM       73 TO 80    SEQ
```

Figure D-11 Single-File Sequential Coordination Match Only: The Application Coding and Report Part 2 (Page 1 of 2)

```
(ER)                                                                                                (S)                                 (ASLGEN'D)    15
    DATALIST: REPORT T.COUNT N.CUSTNO 1.CUSTNO 1.ORDERNO,
                     PF(F.ECORD 1 1) F.MISSPASS T.COMMENT,
              TYPE SUBROUTINE
      TITLE 'SINGLE FILE SEQUENTIAL COORDINATION'
      TITLE 'LIST THE FILE KEYS AND SYSTEM FLAGS'
      TITLE ' '
      TITLE '          <MASTER>  <....CORD 1....>  <BUILDER Fl',
            'AGS>                                            ',
            '                              '
    END REPORT
-------------------------------------------------------------------------------------------------------------------------------------------------------
 ST SUM V  8  PG  PG   I SP  MAXIMUM TB COLUMN  DT  PG  STRT LIN SL RH .......SUBFILE..... C E A S R S N      C  SEQUENCE   LIST
 TY RPT S LPI WID HGT  M FRM LPP PGS TO HEADING POS POS PAGE NUM UB PN ENT    NAME   BLK/ FMT H F G T S U D      O  73 TO 80    SEQ
 MP     P             G           LT TYP POS         NUM      ML TD REC          BUF     C C S L T M S      N
-------------------------------------------------------------------------------------------------------------------------------------------------------
(E1)                                                                                                                                (ASLGEN'D)    16
-------------------------------------------------------------------------------------------------------------------------------------------------------
         SP  Q              E N S D C S T C C M M A G D Q PCT/RAT     P                PARTIAL   SEQUENCE   LIST
STMT SEQ BF  L  FIELD  SEG-  N P E E T U O U N A I V M E L  FIELD   RA- C     PICTURE     STRT NO  73 TO 80   SEQ
TYPE NO. COL F  NAME     LVL D R Q S L B T M T X N G R C F  NAME    TIO T      EDIT      CHR CHR
-------------------------------------------------------------------------------------------------------------------------------------------------------
(R1)        (T,COUNT   )                                                                                                            (ASLGEN'D)    17
(R1)        (N,CUSTNO  ) 001-1                                                                                                      (ASLGEN'D)    18
(R1)        (1,CUSTNO  ) 001-1                                                                                                      (ASLGEN'D)    19
(R1)        (1,ORDERNO ) 001-1                                                                                                      (ASLGEN'D)    20
(R1)        (F,ECORD   )                                                                                                ( 1)( 1)    (ASLGEN'D)    21
(R1)        (F,MISSPASS)                                                                                                            (ASLGEN'D)    22
(R1)        (T,COMMENT )                                                                                                            (ASLGEN'D)    23
-------------------------------------------------------------------------------------------------------------------------------------------------------
STMT  SEQ                                                                                                                           SEQUENCE   LIST
TYPE  NO.                                                  TEXT                                                                     73 TO 80   SEQ
-------------------------------------------------------------------------------------------------------------------------------------------------------
(T1)  (101)                (SINGLE FILE SEQUENTIAL COORDINATION#                    )                                               (ASLGEN'D)    24
(T1)  (102)                (LIST THE FILE KEYS AND SYSTEM FLAGS#                    )                                               (ASLGEN'D)    25
(T1)  (103)                ( #                                                      )                                               (ASLGEN'D)    26
(T1)  (104)                (          <MASTER>  <....CORD 1....>  <BUILDER FlAGS>   )                                               (ASLGEN'D)    27
(T1)  (105)                                                                                                                         (ASLGEN'D)    28
(T1)  (106)                (            #                                           )                                               (ASLGEN'D)    29
NOV 03, 2001  13.23.47                                                                                                                   PAGE     5
** MK4W204  TYPE 0  NUMBER OF MESSAGES PRINTED IS    0.                                                                                  *****
                    M4OLD    -       5 RECORDS INPUT
                    M4CORD1  -       9 RECORDS INPUT
                    M4REPO   -      23 RECORDS OUTPUT
                    M4INPUT  -      30 RECORDS INPUT
                    M4LIST   -     171 RECORDS OUTPUT
                      15 TRACKS ASSIGNED TO M4LIB    --   13 TRACKS NOT FULL. LIBRARY DIRECTORY BLOCKING FACTOR IS  800.
** MK4W209  TYPE 0  1013192 BYTES OF MAIN STORAGE UNUSED DURING DECODING PHASE                                                          *****
                    1014736 BYTES OF MAIN STORAGE UNUSED DURING COMPILATION PHASE
                     565248 BYTES OF MAIN STORAGE UNUSED DURING PROCESSING PHASE
** MK4ED02  TYPE 0  REQUEST-DATALIST  REPORT-1  REQUESTOR ID-                                                                          *****
                                                 SINGLE FILE SEQUENTIAL COORDINATION                                          PAGE 1
                                                 LIST THE FILE KEYS AND SYSTEM FLAGS
          <MASTER>  <....CORD 1....>  <BUILDER FlAGS>
-------------------------------------------------------------------------------------------------------------------------------------------------------
   CYCLE   CUSTOMER  CUSTOMER  ORDER   ECORD  MISSPASS       PROCESS CYCLE
   COUNT    NUMBER    NUMBER   NUMBER                         IN CONTROL
-------------------------------------------------------------------------------------------------------------------------------------------------------
       1   00002     00002    20003     M            SEQUENTIAL COORDINATION
       2   00002     00002    20005     M            SEQUENTIAL COORDINATION
       3   00002       -        -       X      M     STANDARD PROCESSING
       4   00003     00003    30001     M            SEQUENTIAL COORDINATION
       5   00003       -        -       X      M     STANDARD PROCESSING
       6   00005       -        -       X      X     STANDARD PROCESSING
       7   00010     00010    01001     M            SEQUENTIAL COORDINATION
       8   00010     00010    01004     M            SEQUENTIAL COORDINATION
       9   00010       -        -       X      M     STANDARD PROCESSING
      10   00015       -        -       X      X     STANDARD PROCESSING
** MK4FT06  TYPE 1       10 FIELDS WERE PRINTED AS INVALID, MISSING OR OVERFLOW.                                                       *****
** MK4FT03  TYPE 0  END OF REPORT.                                                                                                     *****
** MK4W204  TYPE 0  NUMBER OF MESSAGES PRINTED IS    3.                                                                                *****
                    M4REPO   -      22 RECORDS INPUT
                    M4LIST   -      22 RECORDS OUTPUT
** MK4W209  TYPE 0  1005032 BYTES OF MAIN STORAGE UNUSED DURING REPORTING PHASE                                                        *****
```

Figure D-11 Single-File Sequential Coordination Match Only: The Application Coding and Report Part 2 (Page 2 of 2)

## Single-File Sequential Coordination All Records – Banner Page

```
NOV 03, 2001  13.23.49                                                                                    PAGE      1
                                          ********************************************
                                          *  VISION:Builder   4400  (OS/390 - 14.0)  *
                                          *            COPYRIGHT (C) 2002            *
                                          *  COMPUTER ASSOCIATES INTERNATIONAL, INC. *
                                          ********************************************
VISION:Builder JCR1 14.0 110201                ENABLED FOR IBM LANGUAGE ENVIRONMENT                BUILD STAMP = 101306,12:51:24.
                                          COMLIB  MODIFICATIONS (PTFs, APARs) INSTALLED
                                   201,202,203,204,205,206,207,208,209,210,212,213,214,215,216
=================================================================================================================================
=                                                                                                                               =
=                    I N S T A L L A T I O N   P A R A M E T E R S   ( M 4 P A R A M S , M A R K L I B P )                     =
=                                                                                                                               =
=  SYSTEM DELIMITER:   #          PAGE HEIGHT:        66         M4LIST WIDTH:       132        DEF WIDTH OF PAGE:  0          =
=                                                                                                                               =
=  AUTO GRAND:         N          HEADING CHAR:       -          SUBTITLE REPEAT:    N          INVALID FIELD:      *          =
=                                                                                                                               =
=  MISSING FIELD:      -          NON-EDIT FIELD:     +          PERCENT CHAR:       %          LEFT SEPARATOR:     (          =
=                                                                                                                               =
=  RIGHT SEPARATOR:    )          SINGLE SEPARATOR:   ,          SOURCE SPACING:     1          PRINT MESSAGES:     Y          =
=                                                                                                                               =
=  CONSOLE MESSAGES:   N          M4REPO BLOCKSIZE:   4,096      INPUT I/O BUFFERS:  2          OUTPUT I/O BUFFERS: 1          =
=                                                                                                                               =
=  SNGL-STEP STORAGE:  8,192      SNGL-STEP SORTSIZE: 524,288    DIGIT SELECT CHAR:  9          ZERO SUPPRESS CHAR: Z          =
=                                                                                                                               =
=  CURRENCY CHAR:      $          PLUS CHAR:          +          MINUS CHAR:         -          CHECK PROTECT CHAR: *          =
=                                                                                                                               =
=  DECIMAL CHAR:       .          GROUPING CHAR:      ,          PRIMARY PLOT CHAR:  X          SECONDARY PLOT CHAR:*          =
=                                                                                                                               =
=  FIT PLOT CHAR:      .          HORIZONTAL AXIS:    _          HORIZONTAL HASH:    |          VERTICAL AXIS:      |          =
=                                                                                                                               =
=  VERTICAL HASH:      -          MINUTES/HOUR:       60         SECONDS/MINUTE:     60         TIME DELIMITER:     HH:MM:SS   =
=                                                                                                                               =
=  DATE FORMAT:      MMM DD, YYYY TODAY FORMAT+DELIM: MM/DD/YY   ISDATE DELIMITER:   YYYY-MM-DD JULIAN DELIMITER:   YY.DDD     =
=                                                                                                                               =
=  SORT PROGRAM CODE:  2          MINCORE VALUE:      12 K       ALT M4LIST WIDTH:   264        ALT DEF W/OF PAGE:  0          =
=                                                                                                                               =
=  MAX LINES OF TRACE: 1,024      ITEM TRACKING:      0          SUPPRESS NDS REPT?: N          DEFAULT MAXGETMN:   1,024 K    =
=                                                                                                                               =
=  CONDITION CODE 1:   0          CONDITION CODE 2:   4          CONDITION CODE 3:   0          CONDITION CODE 4:   16         =
=                                                                                                                               =
=================================================================================================================================
```

Figure D-12 Single-File Sequential Coordination All Records Banner Page

## Single-File Sequential Coordination All Records: The Run Control Commands – Part 1 of 2

```
NOV 03, 2001  13.23.49                                                                                                    PAGE      2
-------------------------------------------------------------------------------------------------------------------------------------
   RUN     STMT    FILE    O N S D U T R A B S D S L  RB  R S M R S            START            END              SEQUENCE   LIST
   NAME    TYPE    NAME    L E E I P R E U U R L C S  PL  J O O F S            SEARCH          SEARCH            73 TO 80    SEQ
                           D W Q R D N P D F T M N T  TK  T P P O R
-------------------------------------------------------------------------------------------------------------------------------------
(F152    ) (RC)  (CUSTOMER) (S)          (S)        (T,G)          (R)                                                (ASLGEN'D)     1
     CONTROL NAME F152 TERM LISTGEN
     FILE MASTER INPUT  CUSTOMER
     FILE CORD1 CUSTHIST ALLRECS
     FILE REPORT
-------------------------------------------------------------------------------------------------------------------------------------
   RUN     STMT  LOGICAL  LBL  DTF/DD      KEY       KEY       KEY      QUALIFIER   USER I/O  ARRAY       PASSWORD/  SEQUENCE   LIST
   NAME    TYPE  FILENAME       NAME     FIELD-1   FIELD-2   FIELD-3   ICF CRD DBI   MODULE   QUAL        AUTH.-ID   73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------------
(F152    ) (RF)  (CUSTOMER) (S) (M4OLD   )                                                                           (ASLGEN'D)     2
(F152    ) (RF)  (CUSTHIST) (S) (M4CORD1 )                                  (U)                                      (ASLGEN'D)     3
```

Figure D-13 Single-File Sequential Coordination All Records: The Run Control Commands Part 1

## Single-File Sequential Coordination All Records: The Application Coding and Report – Part 2 of 2

```
=================================================================================================
=                                                                                               =
=                                          F I L E   S U M M A R Y                              =
=                                                                                               =
= JOBNAME: KREJO04X                        STEPNAME: F152                                       =
=                                                                                               =
=  FILE NAME: M4OLD                        DSNAME: BUILDER.R140.CUSTOMER.MASTER.R06#1           =
=  FILE NAME: M4CORD1                      DSNAME: BUILDER.R140.CUSTOMER.HISTORY.R06#1          =
=  FILE NAME: M4LIB                        DSNAME: BUILDER.R140.M4LIB.R06#1                     =
=                                                                                               =
=================================================================================================
NOV 03, 2001  13.23.49                                                                    PAGE     3
                                                 ****************************
                                                 *  PROC NAME - MAIN        *
                                                 *  INPUT STREAM PROCEDURE  *
                                                 ****************************
-------------------------------------------------------------------------------------------------
STMT REPORT                          MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ         BACK BR RNT AB      SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID    ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM      73 TO 80    SEQ
-------------------------------------------------------------------------------------------------
(ER)                                                                      (N)                          (ASLGEN'D)     4
     MAIN: PROC
     ;
     COUNT:   FIELD Z 5  HEADING 'CYCLE' 'COUNT'
     COMMENT: FIELD C 25 HEADING 'PROCESS CYCLE' 'IN CONTROL'
     LET F.ECORD = F.ECORD          ; Make a coordinated procedure
     LET T.COUNT = T.COUNT + 1      ; Count the passes thru the procedure
     IF F.MISSPASS EQ ' ' THEN
       LET T.COMMENT = 'SEQUENTIAL COORDINATION'
     ELSE
       LET T.COMMENT = 'STANDARD PROCESSING'
     END IF
     ;
     CALL REPORT DATALIST           ; Go Report the Data
     ;
     END PROC
     ;
-------------------------------------------------------------------------------------------------
STMT   FIELD    FIELD FLD DEC OUTPT EDIT     INITIAL         LINE 1 OF       LINE 2 OF                  SEQUENCE   LIST
TYPE   NAME     LNGTH TYP PLC EDIT  LGTH      VALUE        COLUMN HEADING  COLUMN HEADING              73 TO 80    SEQ
-------------------------------------------------------------------------------------------------
(TF) (COUNT   ) (  5) (Z)                                (CYCLE        ) (COUNT        )              (ASLGEN'D)     5
(TF) (COMMENT ) ( 25) (C)                                (PROCESS CYCLE ) (IN CONTROL   )              (ASLGEN'D)     6
-------------------------------------------------------------------------------------------------
STMT SEQ  LOG CON .....OPERAND-A.....  OPER  ............OPERAND-B.............   ......RESULT......  PARTIAL FIELD  SEQUENCE   LIST
TYPE NO.  LEV CTR QLF FIELD   SEG-LVL  ATN   QLF FIELD   SEG-LVL  (OR CONSTANT)   QLF FIELD  SEG-LVL  STRT NUMB OPR  73 TO 80    SEQ
-------------------------------------------------------------------------------------------------
(PR) (100)                          (R )   (F,ECORD   )                        (F,ECORD   )                  (ASLGEN'D)     7
(PR) (101)        (T,COUNT   )      (+ )   (D,1                             ) (T,COUNT   )                  (ASLGEN'D)     8
(PR) (102)        (F,MISSPASS)      (EQ)   (C)                                                              (ASLGEN'D)     9
(PR) (103)                          (NS)     (106                          )                                (ASLGEN'D)    10
(PR) (104)                          (R )   (C,SEQUENTIAL COORDINATION      ) (T,COMMENT )                  (ASLGEN'D)    11
(PR) (105)                          (GO)     (107                          )                                (ASLGEN'D)    12
(PR) (106)                          (R )   (C,STANDARD PROCESSING          ) (T,COMMENT )                  (ASLGEN'D)    13
(PR) (107)                          (GO)     (SUB DATALIST                 )                                (ASLGEN'D)    14
NOV 03, 2001  13.23.49                                                                                    PAGE     4
                                                 ****************************
                                                 *  PROC NAME - DATALIST    *
                                                 *  INPUT STREAM PROCEDURE  *
                                                 ****************************
-------------------------------------------------------------------------------------------------
STMT REPORT                          MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ         BACK BR RNT AB      SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID    ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM      73 TO 80    SEQ
-------------------------------------------------------------------------------------------------
(ER)                                                                      (S)                          (ASLGEN'D)    15
     DATALIST: REPORT T.COUNT N.CUSTNO 1.CUSTNO 1.ORDERNO,
                      PF(F.ECORD 1 1) F.MISSPASS T.COMMENT,
                 TYPE SUBROUTINE
       TITLE 'ALL RECORD COORDINATED FILE PROCESSING'
       TITLE 'LIST THE FILE KEYS AND SYSTEM FLAGS'
       TITLE ' '
       TITLE '          <MASTER>  <....CORD 1....>  <BUILDER F1',
             'AGS>                                             ',
             '                                 '
     END REPORT
-------------------------------------------------------------------------------------------------
 ST SUM V  8  PG  PG   I SP  MAXIMUM TB COLUMN  DT  PG  STRT LIN SL RH .......SUBFILE........ C E A S R S N      C  SEQUENCE   LIST
 TY RPT S LPI WID HGT  M FRM LPP PGS TO HEADING POS POS PAGE NUM UB PN ENT    NAME   BLK/ FMT H F G T S U D      O  73 TO 80    SEQ
 MP     P              G            LT TYP POS         NUM      ML TD REC          BUF      C C S L T M S      N
-------------------------------------------------------------------------------------------------
(E1)                                                                                                      (ASLGEN'D)    16
```

Figure D-14 Single-File Sequential Coordination All Records: The Application Coding and Report Part 2 (Page 1 of 2)

```
                SP  Q                    E  N  S  D  C  S  T  C  C  M  M  A  G  D Q PCT/RAT      P                    PARTIAL   SEQUENCE   LIST
STMT SEQ        BF  L  FIELD    SEG-     N  P  E  E  T  U  O  U  N  A  I  V  M  E L  FIELD    RA- C      PICTURE   STRT NO   73 TO 80    SEQ
TYPE NO.        COL F  NAME       LVL    D  R  Q  S  L  B  T  M  T  X  N  G  R  C F  NAME     TIO T       EDIT     CHR CHR
-------------------------------------------------------------------------------------------------------------------------------------------
(R1)          (T,COUNT   )                                                                                                 (ASLGEN'D)    17
(R1)          (N,CUSTNO  ) 001-1                                                                                           (ASLGEN'D)    18
(R1)          (1,CUSTNO  ) 001-1                                                                                           (ASLGEN'D)    19
(R1)          (1,ORDERNO ) 001-1                                                                                           (ASLGEN'D)    20
(R1)          (F,ECORD   )                                                                                    ( 1)( 1)     (ASLGEN'D)    21
(R1)          (F,MISSPASS)                                                                                                 (ASLGEN'D)    22
(R1)          (T,COMMENT )                                                                                                 (ASLGEN'D)    23
-------------------------------------------------------------------------------------------------------------------------------------------
STMT  SEQ                                                                                                                    SEQUENCE   LIST
TYPE  NO.                                                  TEXT                                                              73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------------------
(T1) (101)                (ALL RECORD COORDINATED FILE PROCESSING#                  )                                       (ASLGEN'D)    24
(T1) (102)                (LIST THE FILE KEYS AND SYSTEM FLAGS#                     )                                       (ASLGEN'D)    25
(T1) (103)                ( #                                                       )                                       (ASLGEN'D)    26
(T1) (104)                (          <MASTER>  <....CORD 1....>  <BUILDER FlAGS>    )                                       (ASLGEN'D)    27
(T1) (105)                                                                                                                 (ASLGEN'D)    28
(T1) (106)                (            #                                            )                                       (ASLGEN'D)    29
NOV 03, 2001  13.23.49                                                                                                          PAGE     5
** MK4W204  TYPE 0  NUMBER OF MESSAGES PRINTED IS    0.                                                                                *****
                    M4OLD    -       5 RECORDS INPUT
                    M4CORD1  -       9 RECORDS INPUT
                    M4REPO   -      28 RECORDS OUTPUT
                    M4INPUT  -      30 RECORDS INPUT
                    M4LIST   -     171 RECORDS OUTPUT
                      15 TRACKS ASSIGNED TO M4LIB    --   13 TRACKS NOT FULL. LIBRARY DIRECTORY BLOCKING FACTOR IS  800.
** MK4W209  TYPE 0  1013192 BYTES OF MAIN STORAGE UNUSED DURING DECODING PHASE                                                      *****
                    1014736 BYTES OF MAIN STORAGE UNUSED DURING COMPILATION PHASE
                     565248 BYTES OF MAIN STORAGE UNUSED DURING PROCESSING PHASE
** MK4ED02  TYPE 0  REQUEST-DATALIST  REPORT-1  REQUESTOR ID-                                                                      *****
                                                ALL RECORD COORDINATED FILE PROCESSING                                  PAGE 1
                                                 LIST THE FILE KEYS AND SYSTEM FLAGS
          <MASTER>  <....CORD 1....>  <BUILDER FlAGS>
-------------------------------------------------------------------------------------------------------------------------------------------
   CYCLE   CUSTOMER  CUSTOMER  ORDER   ECORD  MISSPASS        PROCESS CYCLE
   COUNT    NUMBER    NUMBER   NUMBER                          IN CONTROL
-------------------------------------------------------------------------------------------------------------------------------------------
       1   00002     00001    10001     L             SEQUENTIAL COORDINATION
       2   00002     00001    10004     L             SEQUENTIAL COORDINATION
       3   00002     00001    10005     L             SEQUENTIAL COORDINATION
       4   00002     00002    20003     M             SEQUENTIAL COORDINATION
       5   00002     00002    20005     M             SEQUENTIAL COORDINATION
       6   00002       -        -       X       M     STANDARD PROCESSING
       7   00003     00003    30001     M             SEQUENTIAL COORDINATION
       8   00003       -        -       X       M     STANDARD PROCESSING
       9   00005       -        -       X       X     STANDARD PROCESSING
      10   00010     00010    01001     M             SEQUENTIAL COORDINATION
      11   00010     00010    01004     M             SEQUENTIAL COORDINATION
      12   00010       -        -       X       M     STANDARD PROCESSING
      13   00015       -        -       X       X     STANDARD PROCESSING
      14     -       00020    02003     L             SEQUENTIAL COORDINATION
      15     -         -        -       X       X     STANDARD PROCESSING
** MK4FT06  TYPE 1       14 FIELDS WERE PRINTED AS INVALID, MISSING OR OVERFLOW.                                                   *****
** MK4FT03  TYPE 0  END OF REPORT.                                                                                                 *****
** MK4W204  TYPE 0  NUMBER OF MESSAGES PRINTED IS    3.                                                                            *****
                    M4REPO   -      27 RECORDS INPUT
                    M4LIST   -      27 RECORDS OUTPUT
** MK4W209  TYPE 0  1005032 BYTES OF MAIN STORAGE UNUSED DURING REPORTING PHASE                                                    *****
```

Figure D-14 Single-File Sequential Coordination All Records: The Application Coding and Report Part 2 (Page 2 of 2)

## Nested Multi-File Sequential Coordination – Banner Page

```
NOV 03, 2001  13.23.50                                                                                                          PAGE     1

                                                  ******************************************
                                                  *  VISION:Builder   4400  (OS/390 - 14.0)  *
                                                  *           COPYRIGHT (C) 2002             *
                                                  *  COMPUTER ASSOCIATES INTERNATIONAL, INC. *
                                                  ******************************************
VISION:Builder JCR1 14.0 110201                       ENABLED FOR IBM LANGUAGE ENVIRONMENT                  BUILD STAMP = 101306,12:51:24.
                                                   COMLIB  MODIFICATIONS (PTFs, APARs) INSTALLED
                                                201,202,203,204,205,206,207,208,209,210,212,213,214,215,216
=============================================================================================================================================
=                                                                                                                                           =
=                    I N S T A L L A T I O N    P A R A M E T E R S    ( M 4 P A R A M S , M A R K L I B P )                                   =
```

Figure D-15 Nested Multi-File Sequential Coordination - Banner Page (Page 1 of 2)

```
=                                                                                                                                  =
=  SYSTEM DELIMITER:   #          PAGE HEIGHT:        66         M4LIST WIDTH:       132        DEF WIDTH OF PAGE:  0          =
=                                                                                                                                  =
=  AUTO GRAND:         N          HEADING CHAR:       -          SUBTITLE REPEAT:    N          INVALID FIELD:      *          =
=                                                                                                                                  =
=  MISSING FIELD:      -          NON-EDIT FIELD:     +          PERCENT CHAR:       %          LEFT SEPARATOR:     (          =
=                                                                                                                                  =
=  RIGHT SEPARATOR:    )          SINGLE SEPARATOR:   ,          SOURCE SPACING:     1          PRINT MESSAGES:     Y          =
=                                                                                                                                  =
=  CONSOLE MESSAGES:   N          M4REPO BLOCKSIZE:   4,096      INPUT I/O BUFFERS:  2          OUTPUT I/O BUFFERS: 1          =
=                                                                                                                                  =
=  SNGL-STEP STORAGE:  8,192      SNGL-STEP SORTSIZE: 524,288    DIGIT SELECT CHAR:  9          ZERO SUPPRESS CHAR: Z          =
=                                                                                                                                  =
=  CURRENCY CHAR:      $          PLUS CHAR:          +          MINUS CHAR:         -          CHECK PROTECT CHAR: *          =
=                                                                                                                                  =
=  DECIMAL CHAR:       .          GROUPING CHAR:      ,          PRIMARY PLOT CHAR:  X          SECONDARY PLOT CHAR:*          =
=                                                                                                                                  =
=  FIT PLOT CHAR:      .          HORIZONTAL AXIS:    _          HORIZONTAL HASH:    |          VERTICAL AXIS:      |          =
=                                                                                                                                  =
=  VERTICAL HASH:      -          MINUTES/HOUR:       60         SECONDS/MINUTE:     60         TIME DELIMITER:     HH:MM:SS   =
=                                                                                                                                  =
=  DATE FORMAT:      MMM DD, YYYY TODAY FORMAT+DELIM: MM/DD/YY  ISDATE DELIMITER:   YYYY-MM-DD  JULIAN DELIMITER:   YY.DDD     =
=                                                                                                                                  =
=  SORT PROGRAM CODE:  2          MINCORE VALUE:      12 K       ALT M4LIST WIDTH:   264        ALT DEF W/OF PAGE:  0          =
=                                                                                                                                  =
=  MAX LINES OF TRACE: 1,024      ITEM TRACKING:      0          SUPPRESS NDS REPT?: N          DEFAULT MAXGETMN:   1,024 K    =
=                                                                                                                                  =
=  CONDITION CODE 1:   0          CONDITION CODE 2:   4          CONDITION CODE 3:   0          CONDITION CODE 4:   16         =
=                                                                                                                                  =
==================================================================================================================================
```

Figure D-15 Nested Multi-File Sequential Coordination - Banner Page (Page 2 of 2)

## Nested Multi-File Sequential Coordination: Run Control Commands – Part 1 of 2

```
NOV 03, 2001  13.23.50                                                                                                  PAGE      2
-------------------------------------------------------------------------------------------------------------------------------------
   RUN     STMT    FILE    O N S D U T R A B S D S L  RB  R S M R S            START           END              SEQUENCE   LIST
   NAME    TYPE    NAME    L E E I P R E U U R L C S  PL  J O O F S            SEARCH          SEARCH           73 TO 80    SEQ
                           D W Q R D N P D F T M N T  TK  T P P O R
-------------------------------------------------------------------------------------------------------------------------------------
(F155    ) (RC) (CUSTOMER) (S)          (S)        (T,G)          (R)                                          (ASLGEN'D)     1
     CONTROL NAME F155 TERM LISTGEN
     FILE MASTER INPUT  CUSTOMER
     FILE CORD1 CUSTHIST MATCHONLY
     FILE CORD2 ITEMHIST MATCHONLY
     FILE REPORT
-------------------------------------------------------------------------------------------------------------------------------------
   RUN     STMT  LOGICAL  LBL  DTF/DD      KEY       KEY       KEY      QUALIFIER   USER I/O  ARRAY       PASSWORD/  SEQUENCE   LIST
   NAME    TYPE  FILENAME       NAME     FIELD-1   FIELD-2   FIELD-3   ICF CRD DBI   MODULE   QUAL        AUTH.-ID   73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------------
(F155    ) (RF) (CUSTOMER) (S) (M4OLD    )                                                                  (ASLGEN'D)     2
(F155    ) (RF) (CUSTHIST) (S) (M4CORD1 )                                                                   (ASLGEN'D)     3
(F155    ) (RF) (ITEMHIST) (S) (M4CORD2 )                                                                   (ASLGEN'D)     4
```

Figure D-16 Nested Multi-File Sequential Coordination: Run Control Commands Part 1

## Nested Multi-File Sequential Coordination: The Application Coding and Report – Part 2 of 2

```
==================================================================================================================================
=                                                                                                                                =
=                                                  F I L E   S U M M A R Y                                                       =
=                                                                                                                                =
= JOBNAME: KREJO04X                          STEPNAME: F155                                                                      =
=                                                                                                                                =
=  FILE NAME: M4OLD                          DSNAME: BUILDER.R140.CUSTOMER.MASTER.R06#1                                          =
=  FILE NAME: M4CORD1                        DSNAME: BUILDER.R140.CUSTOMER.HISTORY.R06#1                                         =
=  FILE NAME: M4CORD2                        DSNAME: BUILDER.R140.ITEM.HISTORY.R06#1                                             =
=  FILE NAME: M4LIB                          DSNAME: BUILDER.R140.M4LIB.R06#1                                                    =
=                                                                                                                                =
==================================================================================================================================
NOV 03, 2001  13.23.50                                                                                                  PAGE      3
                                                  ******************************
```

Figure D-17 Nested Multi-File Sequential Coordination: The Application Coding and Report Part 2 (Page 1 of 3)

```
                                        *  PROC NAME - MAIN        *
                                        *  INPUT STREAM PROCEDURE  *
                                        ****************************

-------------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                              MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID        ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM       73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------------
(ER)                                                                          (N)                              (ASLGEN'D)     5
    MAIN: PROC
    ;
    COUNT:   FIELD Z 5  HEADING 'CYCLE' 'COUNT'
    COMMENT: FIELD C 25 HEADING 'PROCESS CYCLE' 'IN CONTROL'
    LET F.ECORD = F.ECORD          ; Make a coordinated procedure
    LET T.COUNT = T.COUNT + 1      ; Count the passes thru the procedure
    IF F.MISSPASS EQ ' ' THEN
      LET T.COMMENT = 'SEQUENTIAL COORDINATION'
    ELSE
      LET T.COMMENT = 'STANDARD PROCESSING'
    END IF
    ;
    CALL REPORT DATALIST           ; Go Report the Data
    ;
    END PROC
    ;
-------------------------------------------------------------------------------------------------------------------------------------
STMT   FIELD    FIELD FLD DEC OUTPT EDIT     INITIAL        LINE 1 OF       LINE 2 OF                              SEQUENCE   LIST
TYPE   NAME     LNGTH TYP PLC EDIT  LGTH      VALUE        COLUMN HEADING  COLUMN HEADING                         73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------------
(TF) (COUNT   ) ( 5) (Z)                                  (CYCLE        ) (COUNT        )                        (ASLGEN'D)     6
(TF) (COMMENT ) ( 25) (C)                                 (PROCESS CYCLE ) (IN CONTROL    )                      (ASLGEN'D)     7
-------------------------------------------------------------------------------------------------------------------------------------
STMT SEQ  LOG CON .....OPERAND-A.....  OPER  .............OPERAND-B.............  ......RESULT......  PARTIAL FIELD  SEQUENCE   LIST
TYPE NO.  LEV CTR QLF FIELD   SEG-LVL  AIN   QLF FIELD   SEG-LVL  (OR CONSTANT)  QLF FIELD  SEG-LVL  STRT NUMB OPR  73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------------
(PR) (100)                             (R )  (F,ECORD   )                        (F,ECORD   )                    (ASLGEN'D)     8
(PR) (101)          (T,COUNT   )       (+ )  (D,1                             ) (T,COUNT   )                    (ASLGEN'D)     9
(PR) (102)          (F,MISSPASS)       (EQ)  (C)                                                                (ASLGEN'D)    10
(PR) (103)                             (NS)    (106                           )                                 (ASLGEN'D)    11
(PR) (104)                             (R )  (C,SEQUENTIAL COORDINATION       ) (T,COMMENT )                    (ASLGEN'D)    12
(PR) (105)                             (GO)    (107                           )                                 (ASLGEN'D)    13
(PR) (106)                             (R )  (C,STANDARD PROCESSING           ) (T,COMMENT )                    (ASLGEN'D)    14
(PR) (107)                             (GO)    (SUB DATALIST                  )                                 (ASLGEN'D)    15
NOV 03, 2001  13.23.50                                                                                              PAGE     4
                                        ****************************
                                        *  PROC NAME - DATALIST    *
                                        *  INPUT STREAM PROCEDURE  *
                                        ****************************

-------------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                              MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID        ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM       73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------------
(ER)                                                                          (S)                              (ASLGEN'D)    16
    DATALIST: REPORT T.COUNT N.CUSTNO 1.CUSTNO 1.ORDERNO,
                     2.CUSTNO 2.ORDERNO 2.ITEMNO,
                     PF(F.ECORD 1 2) F.MISSPASS T.COMMENT,
                TYPE SUBROUTINE
      TITLE 'NESTED MULTI-FILE SEQUENTIAL COORDINATION'
      TITLE 'LIST THE FILE KEYS AND SYSTEM FLAGS'
      TITLE ' '
      TITLE '          <MASTER>  <....CORD 1....>  <........CO',
            'RD 2........>   <BUILDER FLAGS>                  ',
            '                                 '
    END REPORT
-------------------------------------------------------------------------------------------------------------------------------------
 ST SUM V  8  PG  PG   I SP  MAXIMUM TB COLUMN  DT  PG  STRT LIN SL RH .......SUBFILE........ C E A S R S N      C  SEQUENCE   LIST
 TY RPT S LPI WID HGT  M FRM LPP PGS TO HEADING POS POS PAGE NUM UB PN ENT    NAME   BLK/ FMT H F G T S U D      O  73 TO 80    SEQ
 MP     P             G            LT TYP POS         NUM      ML TD REC           BUF      C C S L T M S      N
-------------------------------------------------------------------------------------------------------------------------------------
(E1)                                                                                                            (ASLGEN'D)    17
-------------------------------------------------------------------------------------------------------------------------------------
         SP  Q              E  N  S  D  C  S  T  C  C  M  M  A  G  D Q PCT/RAT     P                PARTIAL   SEQUENCE   LIST
STMT SEQ BF  L  FIELD  SEG-  N  P  E  E  T  U  O  U  N  A  I  V  M  E L  FIELD  RA- C     PICTURE    STRT NO  73 TO 80    SEQ
TYPE NO. COL F  NAME     LVL  D  R  Q  S  L  B  T  M  T  X  N  G  R  C F  NAME   TIO T      EDIT      CHR CHR
-------------------------------------------------------------------------------------------------------------------------------------
(R1)         (T,COUNT   )                                                                                       (ASLGEN'D)    18
(R1)         (N,CUSTNO  ) 001-1                                                                                 (ASLGEN'D)    19
(R1)         (1,CUSTNO  ) 001-1                                                                                 (ASLGEN'D)    20
(R1)         (1,ORDERNO ) 001-1                                                                                 (ASLGEN'D)    21
(R1)         (2,CUSTNO  ) 001-1                                                                                 (ASLGEN'D)    22
(R1)         (2,ORDERNO ) 001-1                                                                                 (ASLGEN'D)    23
(R1)         (2,ITEMNO  ) 001-1                                                                                 (ASLGEN'D)    24
(R1)         (F,ECORD   )                                                                          ( 1)( 2)     (ASLGEN'D)    25
(R1)         (F,MISSPASS)                                                                                       (ASLGEN'D)    26
(R1)         (T,COMMENT )                                                                                       (ASLGEN'D)    27
-------------------------------------------------------------------------------------------------------------------------------------
```

Figure D-17 Nested Multi-File Sequential Coordination: The Application Coding and Report Part 2 (Page 2 of 3)

```
STMT  SEQ                                                                                                    SEQUENCE   LIST
TYPE  NO.                                                TEXT                                                73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(T1)  (101)              (NESTED MULTI-FILE SEQUENTIAL COORDINATION#                )                          (ASLGEN'D)    28
(T1)  (102)              (LIST THE FILE KEYS AND SYSTEM FLAGS#                      )                          (ASLGEN'D)    29
(T1)  (103)              ( #                                                        )                          (ASLGEN'D)    30
(T1)  (104)              (          <MASTER>  <....CORD 1....>  <........CORD 2.....)                          (ASLGEN'D)    31
(T1)  (105)              (...>   <BUILDER FLAGS>                                    )                          (ASLGEN'D)    32
(T1)  (106)              (            #                                             )                          (ASLGEN'D)    33
NOV 03, 2001  13.23.50                                                                                             PAGE     5
** MK4W204  TYPE 0  NUMBER OF MESSAGES PRINTED IS    0.                                                                    *****
                    M4OLD    -       5 RECORDS INPUT
                    M4CORD1  -       9 RECORDS INPUT
                    M4CORD2  -      16 RECORDS INPUT
                    M4REPO   -      30 RECORDS OUTPUT
                    M4INPUT  -      32 RECORDS INPUT
                    M4LIST   -     179 RECORDS OUTPUT
                      15 TRACKS ASSIGNED TO M4LIB    --   13 TRACKS NOT FULL. LIBRARY DIRECTORY BLOCKING FACTOR IS  800.
** MK4W209  TYPE 0  1012136 BYTES OF MAIN STORAGE UNUSED DURING DECODING PHASE                                         *****
                    1014056 BYTES OF MAIN STORAGE UNUSED DURING COMPILATION PHASE
                     565248 BYTES OF MAIN STORAGE UNUSED DURING PROCESSING PHASE
** MK4ED02  TYPE 0  REQUEST-DATALIST  REPORT-1  REQUESTOR ID-                                                         *****
                                        NESTED MULTI-FILE SEQUENTIAL COORDINATION                                PAGE 1
                                          LIST THE FILE KEYS AND SYSTEM FLAGS
          <MASTER>  <....CORD 1....>  <........CORD 2........>   <BUILDER FLAGS>
-------------------------------------------------------------------------------------------------------------------------------
   CYCLE   CUSTOMER  CUSTOMER  ORDER   CUSTOMER  ORDER    ITEM    ECORD  MISSPASS        PROCESS CYCLE
   COUNT    NUMBER    NUMBER   NUMBER   NUMBER   NUMBER  NUMBER                          IN CONTROL
-------------------------------------------------------------------------------------------------------------------------------
       1   00002     00002    20003    00002    20003   H11       MM             SEQUENTIAL COORDINATION
       2   00002     00002    20003    00002    20003   H68       MM             SEQUENTIAL COORDINATION
       3   00002     00002    20003    00002    20009   E68       MM             SEQUENTIAL COORDINATION
       4   00002     00002    20005       -        -          -   MX             SEQUENTIAL COORDINATION
       5   00002        -        -       -        -          -   XX       M     STANDARD PROCESSING
       6   00003     00003    30001    00003    30001   E68       MM             SEQUENTIAL COORDINATION
       7   00003        -        -       -        -          -   XX       M     STANDARD PROCESSING
       8   00005        -        -       -        -          -   XX       X     STANDARD PROCESSING
       9   00010     00010    01001    00010    01001   N34       MM             SEQUENTIAL COORDINATION
      10   00010     00010    01001    00010    01004   E68       MM             SEQUENTIAL COORDINATION
      11   00010     00010    01001    00010    01004   M52       MM             SEQUENTIAL COORDINATION
      12   00010     00010    01004       -        -          -   MX             SEQUENTIAL COORDINATION
      13   00010        -        -       -        -          -   XX       M     STANDARD PROCESSING
      14   00015        -        -    00015    15001   O17       XM             SEQUENTIAL COORDINATION
      15   00015        -        -    00015    15006   N11       XM             SEQUENTIAL COORDINATION
      16   00015        -        -    00015    15006   N19       XM             SEQUENTIAL COORDINATION
      17   00015        -        -       -        -          -   XX       M     STANDARD PROCESSING
** MK4FT06  TYPE 1       37 FIELDS WERE PRINTED AS INVALID, MISSING OR OVERFLOW.                                      *****
** MK4FT03  TYPE 0  END OF REPORT.                                                                                     *****
** MK4W204  TYPE 0  NUMBER OF MESSAGES PRINTED IS    3.                                                                *****
                    M4REPO   -      29 RECORDS INPUT
                    M4LIST   -      29 RECORDS OUTPUT
** MK4W209  TYPE 0  1004592 BYTES OF MAIN STORAGE UNUSED DURING REPORTING PHASE                                        *****
```

Figure D-17 Nested Multi-File Sequential Coordination: The Application Coding and Report Part 2 (Page 3 of 3)

## Parallel Multi-File Sequential Coordination Chained to the Master File – Banner Page

```
=                                                                                                                     =
=                  I N S T A L L A T I O N   P A R A M E T E R S   ( M 4 P A R A M S , M A R K L I B P )              =
=                                                                                                                     =
=  SYSTEM DELIMITER:   #          PAGE HEIGHT:        66         M4LIST WIDTH:      132        DEF WIDTH OF PAGE:  0    =
=                                                                                                                     =
=  AUTO GRAND:         N          HEADING CHAR:       -          SUBTITLE REPEAT:   N          INVALID FIELD:      *    =
=                                                                                                                     =
=  MISSING FIELD:      -          NON-EDIT FIELD:     +          PERCENT CHAR:      %          LEFT SEPARATOR:     (    =
=                                                                                                                     =
=  RIGHT SEPARATOR:    )          SINGLE SEPARATOR:   ,          SOURCE SPACING:    1          PRINT MESSAGES:     Y    =
=                                                                                                                     =
=  CONSOLE MESSAGES:   N          M4REPO BLOCKSIZE:   4,096      INPUT I/O BUFFERS: 2          OUTPUT I/O BUFFERS: 1    =
=                                                                                                                     =
=  SNGL-STEP STORAGE:  8,192      SNGL-STEP SORTSIZE: 524,288    DIGIT SELECT CHAR: 9          ZERO SUPPRESS CHAR: Z    =
=                                                                                                                     =
=  CURRENCY CHAR:      $          PLUS CHAR:          +          MINUS CHAR:        -          CHECK PROTECT CHAR: *    =
=                                                                                                                     =
=  DECIMAL CHAR:       .          GROUPING CHAR:      ,          PRIMARY PLOT CHAR: X          SECONDARY PLOT CHAR:*    =
=                                                                                                                     =
=  FIT PLOT CHAR:      .          HORIZONTAL AXIS:    _          HORIZONTAL HASH:   |          VERTICAL AXIS:       |    =
=                                                                                                                     =
```

Figure D-18 Parallel Multi-File Sequential Coordination Chained to the Master File - Banner Page (Page 1 of 2)

```
=  VERTICAL HASH:      -          MINUTES/HOUR:       60         SECONDS/MINUTE:     60         TIME DELIMITER:     HH:MM:SS   =
=                                                                                                                                =
=  DATE FORMAT:      MMM DD, YYYY  TODAY FORMAT+DELIM: MM/DD/YY   ISDATE DELIMITER:   YYYY-MM-DD  JULIAN DELIMITER:   YY.DDD     =
=                                                                                                                                =
=  SORT PROGRAM CODE:  2          MINCORE VALUE:      12 K       ALT M4LIST WIDTH:   264        ALT DEF W/OF PAGE:  0          =
=                                                                                                                                =
=  MAX LINES OF TRACE: 1,024      ITEM TRACKING:      0          SUPPRESS NDS REPT?: N          DEFAULT MAXGETMN:   1,024 K    =
=                                                                                                                                =
=  CONDITION CODE 1:   0          CONDITION CODE 2:   4          CONDITION CODE 3:   0          CONDITION CODE 4:   16         =
=                                                                                                                                =
==================================================================================================================================
```

Figure D-18 Parallel Multi-File Sequential Coordination Chained to the Master File - Banner Page (Page 2 of 2)

## Parallel Multi-File Sequential Coordination Chained to the Master File: Run Control Commands – Part 1 of 2

```
NOV 03, 2001  13.23.51                                                                                                  PAGE      2
-------------------------------------------------------------------------------------------------------------------------------------
   RUN     STMT    FILE    O N S D U T R A B S D S L  RB  R S M R S            START            END              SEQUENCE   LIST
   NAME    TYPE    NAME    L E E I P R E U U R L C S  PL  J O O F S            SEARCH           SEARCH           73 TO 80    SEQ
                           D W Q R D N P D F T M N T  TK  T P P O R
-------------------------------------------------------------------------------------------------------------------------------------
(F156    ) (RC)  (CUSTOMER) (S)          (S)        (T,G)          (R)                                                (ASLGEN'D)     1
     CONTROL NAME F156 TERM LISTGEN
     FILE MASTER INPUT  CUSTOMER
     FILE CORD1 CUSTHIST CHAIN TO M
     FILE CORD2 ITEMHIST CHAIN TO M
     FILE REPORT
-------------------------------------------------------------------------------------------------------------------------------------
   RUN     STMT  LOGICAL  LBL  DTF/DD      KEY       KEY       KEY      QUALIFIER   USER I/O  ARRAY       PASSWORD/  SEQUENCE   LIST
   NAME    TYPE  FILENAME       NAME     FIELD-1   FIELD-2   FIELD-3   ICF CRD DBI   MODULE   QUAL        AUTH.-ID   73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------------
(F156    ) (RF)  (CUSTOMER) (S) (M4OLD   )                                                                          (ASLGEN'D)     2
(F156    ) (RF)  (CUSTHIST) (S) (M4CORD1 )                                   (M)                                   (ASLGEN'D)     3
(F156    ) (RF)  (ITEMHIST) (S) (M4CORD2 )                                   (M)                                   (ASLGEN'D)     4
```

Figure D-19 Parallel Multi-File Sequential Coordination Chained to the Master File: Run Control Commands Part 1

## Parallel Multi-File Sequential Coordination Chained to the Master File: The Application Coding and Report – Part 2 of 2

```
==================================================================================================================================
=                                                                                                                                =
=                                                  F I L E   S U M M A R Y                                                       =
=                                                                                                                                =
= JOBNAME: KREJO04X                          STEPNAME: F156                                                                      =
=                                                                                                                                =
=  FILE NAME: M4OLD                          DSNAME: BUILDER.R140.CUSTOMER.MASTER.R06#1                                          =
=  FILE NAME: M4CORD1                        DSNAME: BUILDER.R140.CUSTOMER.HISTORY.R06#1                                         =
=  FILE NAME: M4CORD2                        DSNAME: BUILDER.R140.ITEM.HISTORY.R06#1                                             =
=  FILE NAME: M4LIB                          DSNAME: BUILDER.R140.M4LIB.R06#1                                                    =
=                                                                                                                                =
==================================================================================================================================
NOV 03, 2001  13.23.51                                                                                                  PAGE      3
                                                  ******************************
                                                  *  PROC NAME - MAIN          *
                                                  *  INPUT STREAM PROCEDURE    *
                                                  ******************************
-------------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                                 MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE           REQUESTOR ID          ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM       73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------------
(ER)                                                                                  (N)                             (ASLGEN'D)     5
     MAIN: PROC
     ;
     COUNT:   FIELD Z 5  HEADING 'CYCLE' 'COUNT'
     COMMENT: FIELD C 25 HEADING 'PROCESS CYCLE' 'IN CONTROL'
     LET F.ECORD = F.ECORD          ; Make a coordinated procedure
     LET T.COUNT = T.COUNT + 1      ; Count the passes thru the procedure
     IF F.MISSPASS EQ ' ' THEN
       LET T.COMMENT = 'SEQUENTIAL COORDINATION'
```

Figure D-20 Parallel Multi-File Sequential Coordination Chained to the Master File: The Application Coding and Report Part 2 (Page 1 of 3)

```
    ELSE
      LET T.COMMENT = 'STANDARD PROCESSING'
    END IF
    ;
    CALL REPORT DATALIST           ; Go Report the Data
    ;
    END PROC
    ;
```

```
STMT   FIELD    FIELD FLD DEC OUTPT EDIT      INITIAL         LINE 1 OF       LINE 2 OF                          SEQUENCE   LIST
TYPE   NAME     LNGTH TYP PLC EDIT  LGTH       VALUE        COLUMN HEADING  COLUMN HEADING                      73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------
(TF) (COUNT   ) ( 5) (Z)                                  (CYCLE        ) (COUNT        )                      (ASLGEN'D)     6
(TF) (COMMENT ) ( 25) (C)                                 (PROCESS CYCLE ) (IN CONTROL    )                      (ASLGEN'D)     7
-------------------------------------------------------------------------------------------------------------------------
STMT SEQ  LOG CON .....OPERAND-A.....  OPER  .............OPERAND-B.............   .....RESULT.....  PARTIAL FIELD  SEQUENCE   LIST
TYPE NO.  LEV CTR QLF FIELD   SEG-LVL  ATN   QLF FIELD   SEG-LVL  (OR CONSTANT)   QLF FIELD  SEG-LVL  STRT NUMB OPR  73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------
(PR) (100)                             (R )   (F,ECORD   )                          (F,ECORD   )                      (ASLGEN'D)     8
(PR) (101)          (T,COUNT   )       (+ )   (D,1                               ) (T,COUNT   )                      (ASLGEN'D)     9
(PR) (102)          (F,MISSPASS)       (EQ)   (C)                                                                    (ASLGEN'D)    10
(PR) (103)                             (NS)     (106                            )                                  (ASLGEN'D)    11
(PR) (104)                             (R )   (C,SEQUENTIAL COORDINATION        ) (T,COMMENT )                      (ASLGEN'D)    12
(PR) (105)                             (GO)     (107                            )                                  (ASLGEN'D)    13
(PR) (106)                             (R )   (C,STANDARD PROCESSING            ) (T,COMMENT )                      (ASLGEN'D)    14
(PR) (107)                             (GO)     (SUB DATALIST                   )                                  (ASLGEN'D)    15
NOV 03, 2001  13.23.51                                                                                                    PAGE     4

                                          ****************************
                                          *  PROC NAME - DATALIST    *
                                          *  INPUT STREAM PROCEDURE  *
                                          ****************************
```

```
STMT REPORT                              MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID        ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM       73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------
(ER)                                                                                    (S)                        (ASLGEN'D)    16
    DATALIST: REPORT T.COUNT N.CUSTNO 1.CUSTNO 1.ORDERNO,
                     2.CUSTNO 2.ORDERNO 2.ITEMNO,
                     PF(F.ECORD 1 2) F.MISSPASS T.COMMENT,
                TYPE SUBROUTINE
      TITLE 'PARALLEL MULTI-FILE SEQUENTIAL COORDINATION'
      TITLE 'LIST THE FILE KEYS AND SYSTEM FLAGS'
      TITLE ' '
      TITLE '          <MASTER>  <....CORD 1....>  <........CO',
            'RD 2........>   <BUILDER FLAGS>                  ',
            '                              '
    END REPORT
```

```
 ST SUM V  8  PG  PG   I SP  MAXIMUM TB COLUMN  DT  PG  STRT LIN SL RH .......SUBFILE..... C E A S R S N      C  SEQUENCE   LIST
 TY RPT S LPI WID HGT  M FRM LPP PGS TO HEADING POS POS PAGE NUM UB PN ENT    NAME   BLK/ FMT H F G T S U D      O  73 TO 80    SEQ
 MP     P            G            LT TYP POS         NUM      ML TD REC           BUF      C C S L T M S      N
-------------------------------------------------------------------------------------------------------------------------
(E1)                                                                                                           (ASLGEN'D)    17
```

```
        SP  Q              E N S D C S T C C M M A G D Q PCT/RAT     P                PARTIAL   SEQUENCE   LIST
STMT SEQ BF  L  FIELD   SEG- N P E E T U O U N A I V M E L FIELD   RA- C     PICTURE     STRT NO   73 TO 80    SEQ
TYPE NO. COL F  NAME     LVL D R Q S L B T M T X N G R C F NAME    TIO T      EDIT      CHR CHR
-------------------------------------------------------------------------------------------------------------------------
(R1)        (T,COUNT   )                                                                                       (ASLGEN'D)    18
(R1)        (N,CUSTNO  ) 001-1                                                                                 (ASLGEN'D)    19
(R1)        (1,CUSTNO  ) 001-1                                                                                 (ASLGEN'D)    20
(R1)        (1,ORDERNO ) 001-1                                                                                 (ASLGEN'D)    21
(R1)        (2,CUSTNO  ) 001-1                                                                                 (ASLGEN'D)    22
(R1)        (2,ORDERNO ) 001-1                                                                                 (ASLGEN'D)    23
(R1)        (2,ITEMNO  ) 001-1                                                                                 (ASLGEN'D)    24
(R1)        (F,ECORD   )                                                                        ( 1)( 2)       (ASLGEN'D)    25
(R1)        (F,MISSPASS)                                                                                       (ASLGEN'D)    26
(R1)        (T,COMMENT )                                                                                       (ASLGEN'D)    27
```

```
STMT  SEQ                                                                                                      SEQUENCE   LIST
TYPE  NO.                                          TEXT                                                        73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------
(T1) (101)              (PARALLEL MULTI-FILE SEQUENTIAL COORDINATION#              )                           (ASLGEN'D)    28
(T1) (102)              (LIST THE FILE KEYS AND SYSTEM FLAGS#                      )                           (ASLGEN'D)    29
(T1) (103)              ( #                                                        )                           (ASLGEN'D)    30
(T1) (104)              (          <MASTER>  <....CORD 1....>  <........CORD 2.....)                           (ASLGEN'D)    31
(T1) (105)              (...>   <BUILDER FLAGS>                                    )                           (ASLGEN'D)    32
(T1) (106)              (             #                                            )                           (ASLGEN'D)    33
NOV 03, 2001  13.23.51                                                                                             PAGE     5
** MK4W204  TYPE 0  NUMBER OF MESSAGES PRINTED IS    0.                                                                *****
                    M4OLD    -       5 RECORDS INPUT
                    M4CORD1  -       9 RECORDS INPUT
                    M4CORD2  -      16 RECORDS INPUT
                    M4REPO   -      28 RECORDS OUTPUT
                    M4INPUT  -      32 RECORDS INPUT
```

Figure D-20 Parallel Multi-File Sequential Coordination Chained to the Master File: The Application Coding and Report Part 2 (Page 2 of 3)

```
                M4LIST   -      179 RECORDS OUTPUT
                  15 TRACKS ASSIGNED TO M4LIB    --   13 TRACKS NOT FULL. LIBRARY DIRECTORY BLOCKING FACTOR IS  800.
** MK4W209  TYPE 0  1012136 BYTES OF MAIN STORAGE UNUSED DURING DECODING PHASE                                    *****
                   1014056 BYTES OF MAIN STORAGE UNUSED DURING COMPILATION PHASE
                    565248 BYTES OF MAIN STORAGE UNUSED DURING PROCESSING PHASE
** MK4ED02  TYPE 0  REQUEST-DATALIST  REPORT-1  REQUESTOR ID-                                                     *****
                                    PARALLEL MULTI-FILE SEQUENTIAL COORDINATION                          PAGE 1
                                       LIST THE FILE KEYS AND SYSTEM FLAGS
          <MASTER>  <....CORD 1....>  <........CORD 2........>  <BUILDER FLAGS>
-------------------------------------------------------------------------------------------------------------------
   CYCLE   CUSTOMER  CUSTOMER  ORDER   CUSTOMER  ORDER   ITEM    ECORD  MISSPASS       PROCESS CYCLE
   COUNT    NUMBER    NUMBER   NUMBER   NUMBER   NUMBER  NUMBER                         IN CONTROL
-------------------------------------------------------------------------------------------------------------------
       1   00002     00002    20003    00002    20003   H11      MM            SEQUENTIAL COORDINATION
       2   00002     00002    20005    00002    20003   H68      MM            SEQUENTIAL COORDINATION
       3   00002       -        -      00002    20009   E68      XM            SEQUENTIAL COORDINATION
       4   00002       -        -        -        -       -      XX      M     STANDARD PROCESSING
       5   00003     00003    30001    00003    30001   E68      MM            SEQUENTIAL COORDINATION
       6   00003       -        -        -        -       -      XX      M     STANDARD PROCESSING
       7   00005       -        -        -        -       -      XX      X     STANDARD PROCESSING
       8   00010     00010    01001    00010    01001   N34      MM            SEQUENTIAL COORDINATION
       9   00010     00010    01004    00010    01004   E68      MM            SEQUENTIAL COORDINATION
      10   00010       -        -      00010    01004   M52      XM            SEQUENTIAL COORDINATION
      11   00010       -        -        -        -       -      XX      M     STANDARD PROCESSING
      12   00015       -        -      00015    15001   O17      XM            SEQUENTIAL COORDINATION
      13   00015       -        -      00015    15006   N11      XM            SEQUENTIAL COORDINATION
      14   00015       -        -      00015    15006   N19      XM            SEQUENTIAL COORDINATION
      15   00015       -        -        -        -       -      XX      M     STANDARD PROCESSING
** MK4FT06  TYPE 1       35 FIELDS WERE PRINTED AS INVALID, MISSING OR OVERFLOW.                                 *****
** MK4FT03  TYPE 0  END OF REPORT.                                                                               *****
** MK4W204  TYPE 0  NUMBER OF MESSAGES PRINTED IS    3.                                                          *****
                   M4REPO   -       27 RECORDS INPUT
                   M4LIST   -       27 RECORDS OUTPUT
** MK4W209  TYPE 0  1004592 BYTES OF MAIN STORAGE UNUSED DURING REPORTING PHASE                                   *****
```

Figure D-20 Parallel Multi-File Sequential Coordination Chained to the Master File: The Application Coding and Report Part 2 (Page 3 of 3)

## Parallel Multi-File Sequential Coordination All Records Processing Matching to the Master File – Banner Page

```
NOV 03, 2001  13.23.52                                                                              PAGE      1

                                        ******************************************
                                        *  VISION:Builder   4400  (OS/390 - 14.0)  *
                                        *           COPYRIGHT (C) 2002             *
                                        *  COMPUTER ASSOCIATES INTERNATIONAL, INC. *
                                        ******************************************
VISION:Builder JCR1 14.0 110201              ENABLED FOR IBM LANGUAGE ENVIRONMENT                BUILD STAMP = 101306,12:51:24.
                                          COMLIB  MODIFICATIONS (PTFs, APARs) INSTALLED
                                    201,202,203,204,205,206,207,208,209,210,212,213,214,215,216
=============================================================================================================
=                                                                                                           =
=                    I N S T A L L A T I O N    P A R A M E T E R S    ( M 4 P A R A M S , M A R K L I B P )   =
=                                                                                                           =
=  SYSTEM DELIMITER:   #          PAGE HEIGHT:       66         M4LIST WIDTH:      132        DEF WIDTH OF PAGE:  0          =
=                                                                                                           =
=  AUTO GRAND:         N          HEADING CHAR:      -          SUBTITLE REPEAT:   N          INVALID FIELD:      *          =
=                                                                                                           =
=  MISSING FIELD:      -          NON-EDIT FIELD:    +          PERCENT CHAR:      %          LEFT SEPARATOR:     (          =
=                                                                                                           =
=  RIGHT SEPARATOR:    )          SINGLE SEPARATOR:  ,          SOURCE SPACING:    1          PRINT MESSAGES:     Y          =
=                                                                                                           =
=  CONSOLE MESSAGES:   N          M4REPO BLOCKSIZE:  4,096      INPUT I/O BUFFERS: 2          OUTPUT I/O BUFFERS: 1          =
=                                                                                                           =
=  SNGL-STEP STORAGE:  8,192      SNGL-STEP SORTSIZE: 524,288   DIGIT SELECT CHAR: 9          ZERO SUPPRESS CHAR: Z          =
=                                                                                                           =
=  CURRENCY CHAR:      $          PLUS CHAR:         +          MINUS CHAR:        -          CHECK PROTECT CHAR: *          =
=                                                                                                           =
=  DECIMAL CHAR:       .          GROUPING CHAR:     ,          PRIMARY PLOT CHAR: X          SECONDARY PLOT CHAR:*          =
=                                                                                                           =
=  FIT PLOT CHAR:      .          HORIZONTAL AXIS:   _          HORIZONTAL HASH:   |          VERTICAL AXIS:      |          =
=                                                                                                           =
=  VERTICAL HASH:      -          MINUTES/HOUR:      60         SECONDS/MINUTE:    60         TIME DELIMITER:     HH:MM:SS   =
=                                                                                                           =
```

Figure D-21 Parallel Multi-File Sequential Coordination All Records Processing Matching to the Master File - Banner Page (Page 1 of 2)

```
=  DATE FORMAT:      MMM DD, YYYY  TODAY FORMAT+DELIM: MM/DD/YY    ISDATE DELIMITER:   YYYY-MM-DD  JULIAN DELIMITER:   YY.DDD     =
=                                                                                                                                  =
=  SORT PROGRAM CODE:  2           MINCORE VALUE:      12 K        ALT M4LIST WIDTH:   264         ALT DEF W/OF PAGE:  0          =
=                                                                                                                                  =
=  MAX LINES OF TRACE: 1,024       ITEM TRACKING:      0           SUPPRESS NDS REPT?: N           DEFAULT MAXGETMN:   1,024 K    =
=                                                                                                                                  =
=  CONDITION CODE 1:   0           CONDITION CODE 2:   4           CONDITION CODE 3:   0           CONDITION CODE 4:   16         =
=                                                                                                                                  =
==================================================================================================================================
```

Figure D-21 Parallel Multi-File Sequential Coordination All Records Processing Matching to the Master File - Banner Page (Page 2 of 2)

## Parallel Multi-File Sequential Coordination All Records Processing Matching to the Master File: Run Control Commands – Part 1 of 2

```
NOV 03, 2001  13.23.52                                                                                                  PAGE      2
-----------------------------------------------------------------------------------------------------------------------------------
   RUN     STMT    FILE    O N S D U T R A B S D S L  RB  R S M R S            START           END              SEQUENCE   LIST
   NAME    TYPE    NAME    L E E I P R E U U R L C S  PL  J O O F S            SEARCH          SEARCH           73 TO 80    SEQ
                           D W Q R D N P D F T M N T  TK  T P P O R
-----------------------------------------------------------------------------------------------------------------------------------
(F157    ) (RC)  (CUSTOMER) (S)          (S)        (T,G)          (R)                                                (ASLGEN'D)     1
     CONTROL NAME F157 TERM LISTGEN
     FILE MASTER INPUT  CUSTOMER
     FILE CORD1 CUSTHIST ALLRECS
     FILE CORD2 ITEMHIST ALLRECS
     FILE REPORT
-----------------------------------------------------------------------------------------------------------------------------------
   RUN     STMT  LOGICAL  LBL  DTF/DD      KEY       KEY       KEY      QUALIFIER   USER I/O  ARRAY       PASSWORD/  SEQUENCE   LIST
   NAME    TYPE  FILENAME       NAME     FIELD-1   FIELD-2   FIELD-3   ICF CRD DBI   MODULE   QUAL        AUTH.-ID   73 TO 80    SEQ
-----------------------------------------------------------------------------------------------------------------------------------
(F157    ) (RF)  (CUSTOMER) (S) (M4OLD    )                                                                          (ASLGEN'D)     2
(F157    ) (RF)  (CUSTHIST) (S) (M4CORD1 )                                  (U)                                     (ASLGEN'D)     3
(F157    ) (RF)  (ITEMHIST) (S) (M4CORD2 )                                  (U)                                     (ASLGEN'D)     4
```

Figure D-22 Parallel Multi-File Sequential Coordination All Records Processing Matching to the Master File: Run Control Commands Part 1

## Parallel Multi-File Sequential Coordination All Records Processing Matching to the Master File: The Application Coding and Report – Part 2 of 2

```
==================================================================================================================================
=                                                                                                                                =
=                                                  F I L E   S U M M A R Y                                                       =
=                                                                                                                                =
= JOBNAME: KREJO04X                          STEPNAME: F157                                                                      =
=                                                                                                                                =
=  FILE NAME: M4OLD                          DSNAME: BUILDER.R140.CUSTOMER.MASTER.R06#1                                          =
=  FILE NAME: M4CORD1                        DSNAME: BUILDER.R140.CUSTOMER.HISTORY.R06#1                                         =
=  FILE NAME: M4CORD2                        DSNAME: BUILDER.R140.ITEM.HISTORY.R06#1                                             =
=  FILE NAME: M4LIB                          DSNAME: BUILDER.R140.M4LIB.R06#1                                                    =
=                                                                                                                                =
==================================================================================================================================
NOV 03, 2001  13.23.52                                                                                                  PAGE      3
                                                  ******************************
                                                  *  PROC NAME - MAIN          *
                                                  *  INPUT STREAM PROCEDURE    *
                                                  ******************************
-----------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                                 MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE           REQUESTOR ID          ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM       73 TO 80    SEQ
-----------------------------------------------------------------------------------------------------------------------------------
(ER)                                                                                  (N)                            (ASLGEN'D)     5
     MAIN: PROC
     ;
     COUNT:   FIELD Z 5  HEADING 'CYCLE' 'COUNT'
     COMMENT: FIELD C 25 HEADING 'PROCESS CYCLE' 'IN CONTROL'
     LET F.ECORD = F.ECORD          ; Make a coordinated procedure
     LET T.COUNT = T.COUNT + 1      ; Count the passes thru the procedure
     IF F.MISSPASS EQ ' ' THEN
       LET T.COMMENT = 'SEQUENTIAL COORDINATION'
```

Figure D-23 Parallel Multi-File Sequential Coordination All Records Processing Matching to the Master File: The Application Coding and Report Part 2 (Page 1 of 3)

```
    ELSE
      LET T.COMMENT = 'STANDARD PROCESSING'
    END IF
    ;
    CALL REPORT DATALIST           ; Go Report the Data
    ;
    END PROC
    ;
-------------------------------------------------------------------------------------------------------------------------------
STMT   FIELD    FIELD FLD DEC OUTPT EDIT      INITIAL         LINE 1 OF       LINE 2 OF                          SEQUENCE   LIST
TYPE   NAME     LNGTH TYP PLC EDIT  LGTH       VALUE        COLUMN HEADING  COLUMN HEADING                      73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(TF) (COUNT   ) ( 5) (Z)                                  (CYCLE         ) (COUNT         )                      (ASLGEN'D)     6
(TF) (COMMENT ) ( 25) (C)                                 (PROCESS CYCLE ) (IN CONTROL    )                      (ASLGEN'D)     7
-------------------------------------------------------------------------------------------------------------------------------
STMT SEQ  LOG CON .....OPERAND-A.....  OPER  .............OPERAND-B.............   ......RESULT......  PARTIAL FIELD  SEQUENCE   LIST
TYPE NO.  LEV CTR QLF FIELD   SEG-LVL  ATN   QLF FIELD   SEG-LVL  (OR CONSTANT)   QLF FIELD  SEG-LVL  STRT NUMB OPR  73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(PR) (100)                             (R )  (F,ECORD   )                          (F,ECORD   )                      (ASLGEN'D)     8
(PR) (101)          (T,COUNT   )       (+ )  (D,1                               ) (T,COUNT   )                      (ASLGEN'D)     9
(PR) (102)          (F,MISSPASS)       (EQ)  (C)                                                                    (ASLGEN'D)    10
(PR) (103)                             (NS)    (106                             )                                  (ASLGEN'D)    11
(PR) (104)                             (R )  (C,SEQUENTIAL COORDINATION         ) (T,COMMENT )                     (ASLGEN'D)    12
(PR) (105)                             (GO)    (107                             )                                  (ASLGEN'D)    13
(PR) (106)                             (R )  (C,STANDARD PROCESSING             ) (T,COMMENT )                     (ASLGEN'D)    14
(PR) (107)                             (GO)    (SUB DATALIST                    )                                  (ASLGEN'D)    15
NOV 03, 2001  13.23.52                                                                                                   PAGE     4

                                                ******************************
                                                *  PROC NAME - DATALIST      *
                                                *  INPUT STREAM PROCEDURE    *
                                                ******************************

-------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                              MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID        ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM       73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(ER)                                                                             (S)                                (ASLGEN'D)    16
    DATALIST: REPORT T.COUNT N.CUSTNO 1.CUSTNO 1.ORDERNO,
                     2.CUSTNO 2.ORDERNO 2.ITEMNO,
                     PF(F.ECORD 1 2) F.MISSPASS T.COMMENT,
                TYPE SUBROUTINE
      TITLE 'PARALLEL MULTI-FILE SEQUENTIAL COORDINATION - ALL RECORDS'
      TITLE 'LIST THE FILE KEYS AND SYSTEM FLAGS'
      TITLE ' '
      TITLE '          <MASTER>  <....CORD 1....>  <.........CO',
            'RD 2.........>   <BUILDER FLAGS>                  ',
            '                                 '
    END REPORT
-------------------------------------------------------------------------------------------------------------------------------
 ST SUM V  8  PG  PG   I SP  MAXIMUM TB COLUMN  DT  PG  STRT LIN SL RH .......SUBFILE........ C E A S R S N      C  SEQUENCE   LIST
 TY RPT S LPI WID HGT  M FRM LPP PGS TO HEADING POS POS PAGE NUM UB PN ENT    NAME   BLK/ FMT H F G T S U D      O  73 TO 80    SEQ
 MP     P            G            LT TYP POS         NUM      ML TD REC           BUF      C C S L T M S      N
-------------------------------------------------------------------------------------------------------------------------------
(E1)                                                                                                                (ASLGEN'D)    17
-------------------------------------------------------------------------------------------------------------------------------
         SP  Q               E  N  S  D  C  S  T  C  C  M  M  A  G  D Q PCT/RAT     P                PARTIAL   SEQUENCE   LIST
STMT SEQ BF  L  FIELD  SEG-  N  P  E  E  T  U  O  U  N  A  I  V  M  E L  FIELD   RA- C      PICTURE     STRT NO   73 TO 80    SEQ
TYPE NO. COL F  NAME     LVL  D  R  Q  S  L  B  T  M  T  X  N  G  R  C F  NAME    TIO T       EDIT       CHR CHR
-------------------------------------------------------------------------------------------------------------------------------
(R1)         (T,COUNT   )                                                                                           (ASLGEN'D)    18
(R1)         (N,CUSTNO  ) 001-1                                                                                     (ASLGEN'D)    19
(R1)         (1,CUSTNO  ) 001-1                                                                                     (ASLGEN'D)    20
(R1)         (1,ORDERNO ) 001-1                                                                                     (ASLGEN'D)    21
(R1)         (2,CUSTNO  ) 001-1                                                                                     (ASLGEN'D)    22
(R1)         (2,ORDERNO ) 001-1                                                                                     (ASLGEN'D)    23
(R1)         (2,ITEMNO  ) 001-1                                                                                     (ASLGEN'D)    24
(R1)         (F,ECORD   )                                                                              ( 1)( 2)     (ASLGEN'D)    25
(R1)         (F,MISSPASS)                                                                                           (ASLGEN'D)    26
(R1)         (T,COMMENT )                                                                                           (ASLGEN'D)    27
-------------------------------------------------------------------------------------------------------------------------------
STMT  SEQ                                                                                                           SEQUENCE   LIST
TYPE  NO.                                          TEXT                                                             73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(T1) (101)              (PARALLEL MULTI-FILE SEQUENTIAL COORDINATION - ALL RECORDS# )                                (ASLGEN'D)    28
(T1) (102)              (LIST THE FILE KEYS AND SYSTEM FLAGS#                       )                                (ASLGEN'D)    29
(T1) (103)              ( #                                                         )                                (ASLGEN'D)    30
(T1) (104)              (          <MASTER>  <....CORD 1....>  <.........CORD 2.....)                                (ASLGEN'D)    31
(T1) (105)              (...>   <BUILDER FLAGS>                                     )                                (ASLGEN'D)    32
(T1) (106)              (             #                                             )                                (ASLGEN'D)    33
NOV 03, 2001  13.23.52                                                                                                   PAGE     5
** MK4W204  TYPE 0  NUMBER OF MESSAGES PRINTED IS    0.                                                                  *****
                    M4OLD    -       5 RECORDS INPUT
                    M4CORD1  -       9 RECORDS INPUT
                    M4CORD2  -      16 RECORDS INPUT
```

Figure D-23 Parallel Multi-File Sequential Coordination All Records Processing Matching to the Master File: The Application Coding and Report Part 2 (Page 2 of 3)

```
                M4REPO   -       35 RECORDS OUTPUT
                M4INPUT  -       32 RECORDS INPUT
                M4LIST   -      179 RECORDS OUTPUT
                  15 TRACKS ASSIGNED TO M4LIB    --   13 TRACKS NOT FULL. LIBRARY DIRECTORY BLOCKING FACTOR IS  800.
** MK4W209  TYPE 0  1012136 BYTES OF MAIN STORAGE UNUSED DURING DECODING PHASE                                      *****
                   1014056 BYTES OF MAIN STORAGE UNUSED DURING COMPILATION PHASE
                    565248 BYTES OF MAIN STORAGE UNUSED DURING PROCESSING PHASE
** MK4ED02  TYPE 0  REQUEST-DATALIST  REPORT-1  REQUESTOR ID-                                                       *****
                                  PARALLEL MULTI-FILE SEQUENTIAL COORDINATION - ALL RECORDS                    PAGE 1
                                              LIST THE FILE KEYS AND SYSTEM FLAGS
          <MASTER>  <....CORD 1....>  <........CORD 2........>   <BUILDER FLAGS>
-------------------------------------------------------------------------------------------------------------------------

   CYCLE   CUSTOMER  CUSTOMER  ORDER   CUSTOMER  ORDER    ITEM    ECORD  MISSPASS        PROCESS CYCLE
   COUNT    NUMBER    NUMBER   NUMBER   NUMBER   NUMBER  NUMBER                          IN CONTROL
-------------------------------------------------------------------------------------------------------------------------

       1   00002     00001    10001    00001    10001   M55       LL             SEQUENTIAL COORDINATION
       2   00002       -        -      00001    10001   N94       XL             SEQUENTIAL COORDINATION
       3   00002     00001    10004    00001    10004   M52       LL             SEQUENTIAL COORDINATION
       4   00002       -        -      00001    10004   O17       XL             SEQUENTIAL COORDINATION
       5   00002     00001    10005    00001    10005   O13       LL             SEQUENTIAL COORDINATION
       6   00002     00002    20003    00002    20003   H11       MM             SEQUENTIAL COORDINATION
       7   00002     00002    20005    00002    20003   H68       MM             SEQUENTIAL COORDINATION
       8   00002       -        -      00002    20009   E68       XM             SEQUENTIAL COORDINATION
       9   00002       -        -        -        -         -     XX       M     STANDARD PROCESSING
      10   00003     00003    30001    00003    30001   E68       MM             SEQUENTIAL COORDINATION
      11   00003       -        -        -        -         -     XX       M     STANDARD PROCESSING
      12   00005       -        -        -        -         -     XX       X     STANDARD PROCESSING
      13   00010     00010    01001    00010    01001   N34       MM             SEQUENTIAL COORDINATION
      14   00010     00010    01004    00010    01004   E68       MM             SEQUENTIAL COORDINATION
      15   00010       -        -      00010    01004   M52       XM             SEQUENTIAL COORDINATION
      16   00010       -        -        -        -         -     XX       M     STANDARD PROCESSING
      17   00015       -        -      00015    15001   O17       XM             SEQUENTIAL COORDINATION
      18   00015       -        -      00015    15006   N11       XM             SEQUENTIAL COORDINATION
      19   00015       -        -      00015    15006   N19       XM             SEQUENTIAL COORDINATION
      20   00015       -        -        -        -         -     XX       M     STANDARD PROCESSING
      21     -       00020    02003    00020    02003   E13       LL             SEQUENTIAL COORDINATION
      22     -         -        -        -        -         -     XX       X     STANDARD PROCESSING
** MK4FT06  TYPE 1       46 FIELDS WERE PRINTED AS INVALID, MISSING OR OVERFLOW.                                   *****
** MK4FT03  TYPE 0  END OF REPORT.                                                                                  *****
** MK4W204  TYPE 0  NUMBER OF MESSAGES PRINTED IS    3.                                                            *****
                M4REPO   -       34 RECORDS INPUT
                M4LIST   -       34 RECORDS OUTPUT
** MK4W209  TYPE 0  1004576 BYTES OF MAIN STORAGE UNUSED DURING REPORTING PHASE                                     *****
```

Figure D-23 Parallel Multi-File Sequential Coordination All Records Processing Matching to the Master File: The Application Coding and Report Part 2 (Page 3 of 3)

## Chained Multi-File Sequential Coordination – Banner Page

```
NOV 03, 2001  13.23.54                                                                                    PAGE      1

                                        ******************************************
                                        *  VISION:Builder   4400  (OS/390 - 14.0)  *
                                        *           COPYRIGHT (C) 2002             *
                                        *  COMPUTER ASSOCIATES INTERNATIONAL, INC. *
                                        ******************************************

VISION:Builder JCR1 14.0 110201              ENABLED FOR IBM LANGUAGE ENVIRONMENT                BUILD STAMP = 101306,12:51:24.
                                           COMLIB  MODIFICATIONS (PTFs, APARs) INSTALLED
                                      201,202,203,204,205,206,207,208,209,210,212,213,214,215,216

=                                                                                                                      =
=                  I N S T A L L A T I O N    P A R A M E T E R S    ( M 4 P A R A M S , M A R K L I B P )           =
=                                                                                                                      =
=  SYSTEM DELIMITER:   #          PAGE HEIGHT:       66         M4LIST WIDTH:      132        DEF WIDTH OF PAGE:  0     =
=                                                                                                                      =
=  AUTO GRAND:         N          HEADING CHAR:      -          SUBTITLE REPEAT:   N          INVALID FIELD:      *     =
=                                                                                                                      =
=  MISSING FIELD:      -          NON-EDIT FIELD:    +          PERCENT CHAR:      %          LEFT SEPARATOR:     (     =
=                                                                                                                      =
=  RIGHT SEPARATOR:    )          SINGLE SEPARATOR:  ,          SOURCE SPACING:    1          PRINT MESSAGES:     Y     =
=                                                                                                                      =
=  CONSOLE MESSAGES:   N          M4REPO BLOCKSIZE:  4,096      INPUT I/O BUFFERS: 2          OUTPUT I/O BUFFERS: 1     =
=                                                                                                                      =
=  SNGL-STEP STORAGE:  8,192      SNGL-STEP SORTSIZE: 524,288   DIGIT SELECT CHAR: 9          ZERO SUPPRESS CHAR: Z     =
=                                                                                                                      =
=  CURRENCY CHAR:      $          PLUS CHAR:         +          MINUS CHAR:        -          CHECK PROTECT CHAR: *     =
=                                                                                                                      =
=  DECIMAL CHAR:       .          GROUPING CHAR:     ,          PRIMARY PLOT CHAR: X          SECONDARY PLOT CHAR:*     =
=                                                                                                                      =
```

Figure D-24 Chained Multi-File Sequential Coordination - Banner Page (Page 1 of 2)

```
=  FIT PLOT CHAR:      .          HORIZONTAL AXIS:    _          HORIZONTAL HASH:    |          VERTICAL AXIS:      |          =
=                                                                                                                                =
=  VERTICAL HASH:      -          MINUTES/HOUR:       60         SECONDS/MINUTE:     60         TIME DELIMITER:     HH:MM:SS   =
=                                                                                                                                =
=  DATE FORMAT:      MMM DD, YYYY  TODAY FORMAT+DELIM: MM/DD/YY   ISDATE DELIMITER:   YYYY-MM-DD  JULIAN DELIMITER:   YY.DDD     =
=                                                                                                                                =
=  SORT PROGRAM CODE:  2          MINCORE VALUE:      12 K       ALT M4LIST WIDTH:   264        ALT DEF W/OF PAGE:  0          =
=                                                                                                                                =
=  MAX LINES OF TRACE: 1,024      ITEM TRACKING:      0          SUPPRESS NDS REPT?: N          DEFAULT MAXGETMN:   1,024 K    =
=                                                                                                                                =
=  CONDITION CODE 1:   0          CONDITION CODE 2:   4          CONDITION CODE 3:   0          CONDITION CODE 4:   16         =
=                                                                                                                                =
==================================================================================================================================
```

Figure D-24 Chained Multi-File Sequential Coordination - Banner Page (Page 2 of 2)

## Chained Multi-File Sequential Coordination: Run Control Commands – Part 1 of 2

```
NOV 03, 2001  13.23.54                                                                                                  PAGE      2
-------------------------------------------------------------------------------------------------------------------------------------
   RUN     STMT    FILE    O N S D U T R A B S D S L  RB  R S M R S            START            END              SEQUENCE   LIST
   NAME    TYPE    NAME    L E E I P R E U U R L C S  PL  J O O F S            SEARCH           SEARCH           73 TO 80    SEQ
                           D W Q R D N P D F T M N T  TK  T P P O R
-------------------------------------------------------------------------------------------------------------------------------------
(F159    ) (RC)  (CUSTOMER) (S)         (S)       (T,G)          (R)                                                (ASLGEN'D)     1


     CONTROL NAME F159 TERM LISTGEN
     FILE MASTER INPUT  CUSTOMER
     FILE CORD1 CUSTHIST CHAIN TO M
     FILE CORD2 CUSTHIST CHAIN TO 1
     FILE CORD3 ITEMHIST CHAIN TO 1
     FILE REPORT
-------------------------------------------------------------------------------------------------------------------------------------
   RUN     STMT  LOGICAL  LBL  DTF/DD      KEY       KEY       KEY      QUALIFIER   USER I/O  ARRAY       PASSWORD/  SEQUENCE   LIST
   NAME    TYPE  FILENAME       NAME     FIELD-1   FIELD-2   FIELD-3   ICF CRD DBI   MODULE   QUAL        AUTH.-ID   73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------------
(F159    ) (RF)  (CUSTOMER) (S) (M4OLD   )                                                                          (ASLGEN'D)     2
(F159    ) (RF)  (CUSTHIST) (S) (M4CORD1 )                                  (M)                                     (ASLGEN'D)     3
(F159    ) (RF)  (CUSTHIST) (S) (M4CORD2 )                                  (1)                                     (ASLGEN'D)     4
(F159    ) (RF)  (ITEMHIST) (S) (M4CORD3 )                                  (1)                                     (ASLGEN'D)     5
```

Figure D-25 Chained Multi-File Sequential Coordination: Run Control Commands Part 1

## Chained Multi-File Sequential Coordination: The Application Coding and Report – Part 2 of 2

```
==================================================================================================================================
=                                                                                                                                =
=                                                  F I L E   S U M M A R Y                                                       =
=                                                                                                                                =
= JOBNAME: KREJO04X                          STEPNAME: F159                                                                      =
=                                                                                                                                =
=  FILE NAME: M4OLD                          DSNAME: BUILDER.R140.CUSTOMER.MASTER.R06#1                                          =
=  FILE NAME: M4CORD1                        DSNAME: BUILDER.R140.CUSTOMER.HISTORY.R06#1                                         =
=  FILE NAME: M4CORD2                        DSNAME: BUILDER.R140.CUSTOMER.HISTORY.R06#1                                         =
=  FILE NAME: M4CORD3                        DSNAME: BUILDER.R140.ITEM.HISTORY.R06#1                                             =
=  FILE NAME: M4LIB                          DSNAME: BUILDER.R140.M4LIB.R06#1                                                    =
=                                                                                                                                =
==================================================================================================================================
NOV 03, 2001  13.23.54                                                                                                  PAGE      3
                                                  ******************************
                                                  *  PROC NAME - MAIN          *
                                                  *  INPUT STREAM PROCEDURE    *
                                                  ******************************
-------------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                                 MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE           REQUESTOR ID          ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM       73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------------
(ER)                                                                                (N)                              (ASLGEN'D)     6
     MAIN: PROC
     ;
     COUNT:   FIELD Z 5  HEADING 'CYCLE' 'COUNT'
     COMMENT: FIELD C 25 HEADING 'PROCESS CYCLE' 'IN CONTROL'
```

Figure D-26 Chained Multi-File Sequential Coordination: The Application Coding and Report Part 2 (Page 1 of 3)

```
    LET F.ECORD = F.ECORD          ; Make a coordinated procedure
    LET T.COUNT = T.COUNT + 1      ; Count the passes thru the procedure
    IF F.MISSPASS EQ ' ' THEN
      LET T.COMMENT = 'SEQUENTIAL COORDINATION'
    ELSE
      LET T.COMMENT = 'STANDARD PROCESSING'
    END IF
    ;
    CALL REPORT DATALIST           ; Go Report the Data
    ;
    END PROC
    ;
```

```
STMT   FIELD    FIELD FLD DEC OUTPT EDIT      INITIAL         LINE 1 OF       LINE 2 OF                        SEQUENCE   LIST
TYPE   NAME     LNGTH TYP PLC EDIT  LGTH      VALUE         COLUMN HEADING  COLUMN HEADING                   73 TO 80   SEQ
-------------------------------------------------------------------------------------------------------------------------
(TF) (COUNT   ) (  5) (Z)                                  (CYCLE         ) (COUNT         )                  (ASLGEN'D)     7
(TF) (COMMENT ) ( 25) (C)                                  (PROCESS CYCLE ) (IN CONTROL    )                  (ASLGEN'D)     8
-------------------------------------------------------------------------------------------------------------------------
STMT SEQ  LOG CON .....OPERAND-A.....  OPER  .............OPERAND-B.............   .......RESULT......  PARTIAL FIELD  SEQUENCE   LIST
TYPE NO.  LEV CTR QLF FIELD   SEG-LVL  ATN   QLF FIELD   SEG-LVL  (OR CONSTANT)   QLF FIELD  SEG-LVL  STRT NUMB OPR  73 TO 80   SEQ
-------------------------------------------------------------------------------------------------------------------------
(PR) (100)                             (R )  (F,ECORD   )                          (F,ECORD   )                   (ASLGEN'D)     9
(PR) (101)         (T,COUNT   )        (+ )  (D,1                               ) (T,COUNT   )                   (ASLGEN'D)    10
(PR) (102)         (F,MISSPASS)        (EQ)  (C)                                                                 (ASLGEN'D)    11
(PR) (103)                             (NS)    (106                             )                               (ASLGEN'D)    12
(PR) (104)                             (R )  (C,SEQUENTIAL COORDINATION        ) (T,COMMENT )                   (ASLGEN'D)    13
(PR) (105)                             (GO)    (107                             )                               (ASLGEN'D)    14
(PR) (106)                             (R )  (C,STANDARD PROCESSING            ) (T,COMMENT )                   (ASLGEN'D)    15
(PR) (107)                             (GO)    (SUB DATALIST                    )                               (ASLGEN'D)    16
NOV 03, 2001  13.23.54                                                                                             PAGE     4

                                                ****************************
                                                *  PROC NAME - DATALIST    *
                                                *  INPUT STREAM PROCEDURE  *
                                                ****************************
```

```
STMT REPORT                              MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB      SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID         ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM     73 TO 80   SEQ
-------------------------------------------------------------------------------------------------------------------------
(ER)                                                                            (S)                              (ASLGEN'D)    17
    DATALIST: REPORT T.COUNT N.CUSTNO 1.CUSTNO 1.ORDERNO,
                     2.CUSTNO 2.ORDERNO,
                     3.CUSTNO 3.ORDERNO 3.ITEMNO,
                     PF(F.ECORD 1 3) F.MISSPASS T.COMMENT,
              TYPE SUBROUTINE
      TITLE 'CHAINED MULTI-FILE SEQUENTIAL COORDINATION'
      TITLE 'USING BOTH NESTED AND PARALLEL COORDINATION'
      TITLE 'LIST THE FILE KEYS AND SYSTEM FLAGS'
      TITLE ' '
      TITLE '          <MASTER>  <....CORD 1....>  <....CORD 2',
            '....>  <.........CORD 3.........>   <BUILDER FLAGS> ',
            '                                  '
    END REPORT
```

```
 ST SUM V  8  PG  PG   I SP  MAXIMUM TB COLUMN  DT  PG  STRT LIN SL RH .......SUBFILE........ C E A S R S N      C  SEQUENCE   LIST
 TY RPT S LPI WID HGT  M FRM LPP PGS TO HEADING POS POS PAGE NUM UB PN ENT    NAME   BLK/ FMT H F G T S U D      O  73 TO 80   SEQ
 MP     P            G           LT TYP POS         NUM      ML TD REC          BUF      C C S L T M S      N
-------------------------------------------------------------------------------------------------------------------------
(E1)                                                                                                          (ASLGEN'D)    18
-------------------------------------------------------------------------------------------------------------------------
        SP  Q              E N S D C S T C C M M A G D Q PCT/RAT     P                PARTIAL   SEQUENCE   LIST
STMT SEQ BF  L  FIELD   SEG- N P E E T U O U N A I V M E L FIELD   RA- C     PICTURE     STRT NO  73 TO 80   SEQ
TYPE NO. COL F  NAME     LVL D R Q S L B T M T X N G R C F NAME    TIO T      EDIT      CHR CHR
-------------------------------------------------------------------------------------------------------------------------
(R1)        (T,COUNT   )                                                                                      (ASLGEN'D)    19
(R1)        (N,CUSTNO  ) 001-1                                                                                (ASLGEN'D)    20
(R1)        (1,CUSTNO  ) 001-1                                                                                (ASLGEN'D)    21
(R1)        (1,ORDERNO ) 001-1                                                                                (ASLGEN'D)    22
(R1)        (2,CUSTNO  ) 001-1                                                                                (ASLGEN'D)    23
(R1)        (2,ORDERNO ) 001-1                                                                                (ASLGEN'D)    24
(R1)        (3,CUSTNO  ) 001-1                                                                                (ASLGEN'D)    25
(R1)        (3,ORDERNO ) 001-1                                                                                (ASLGEN'D)    26
(R1)        (3,ITEMNO  ) 001-1                                                                                (ASLGEN'D)    27
(R1)        (F,ECORD   )                                                                        ( 1)( 3)      (ASLGEN'D)    28
(R1)        (F,MISSPASS)                                                                                      (ASLGEN'D)    29
(R1)        (T,COMMENT )                                                                                      (ASLGEN'D)    30
```

```
STMT  SEQ                                                                                                      SEQUENCE   LIST
TYPE  NO.                                          TEXT                                                        73 TO 80   SEQ
-------------------------------------------------------------------------------------------------------------------------
(T1) (101)          (CHAINED MULTI-FILE SEQUENTIAL COORDINATION#                )                              (ASLGEN'D)    31
(T1) (102)          (USING BOTH NESTED AND PARALLEL COORDINATION#                )                              (ASLGEN'D)    32
(T1) (103)          (LIST THE FILE KEYS AND SYSTEM FLAGS#                       )                              (ASLGEN'D)    33
(T1) (104)          ( #                                                         )                              (ASLGEN'D)    34
(T1) (105)          (          <MASTER>  <....CORD 1....>  <....CORD 2....>  <.)                               (ASLGEN'D)    35
```

Figure D-26 Chained Multi-File Sequential Coordination: The Application Coding and Report Part 2 (Page 2 of 3)

```
(T1) (106)                (.......CORD 3........>   <BUILDER FLAGS>                )                    (ASLGEN'D)    36
(T1) (107)                (             #                                         )                    (ASLGEN'D)    37
NOV 03, 2001  13.23.54                                                                                       PAGE      5
** MK4W204  TYPE 0  NUMBER OF MESSAGES PRINTED IS    0.                                                              *****
                    M4OLD    -       5 RECORDS INPUT
                    M4CORD1  -       9 RECORDS INPUT
                    M4CORD2  -       9 RECORDS INPUT
                    M4CORD3  -      13 RECORDS INPUT
                    M4REPO   -      30 RECORDS OUTPUT
                    M4INPUT  -      35 RECORDS INPUT
                    M4LIST   -     188 RECORDS OUTPUT
                      15 TRACKS ASSIGNED TO M4LIB    --   13 TRACKS NOT FULL. LIBRARY DIRECTORY BLOCKING FACTOR IS  800.
** MK4W209  TYPE 0  1011176 BYTES OF MAIN STORAGE UNUSED DURING DECODING PHASE                                      *****
                    1013408 BYTES OF MAIN STORAGE UNUSED DURING COMPILATION PHASE
                     565248 BYTES OF MAIN STORAGE UNUSED DURING PROCESSING PHASE
** MK4ED02  TYPE 0  REQUEST-DATALIST  REPORT-1  REQUESTOR ID-                                                      *****
                                    CHAINED MULTI-FILE SEQUENTIAL COORDINATION                                PAGE 1
                                   USING BOTH NESTED AND PARALLEL COORDINATION
                                      LIST THE FILE KEYS AND SYSTEM FLAGS
          <MASTER>  <....CORD 1....>  <....CORD 2....>  <........CORD 3........>   <BUILDER FLAGS>
-------------------------------------------------------------------------------------------------------------------------
   CYCLE   CUSTOMER  CUSTOMER  ORDER   CUSTOMER  ORDER   CUSTOMER  ORDER    ITEM    ECORD  MISSPASS        PROCESS CYCLE
   COUNT    NUMBER    NUMBER  NUMBER   NUMBER  NUMBER    NUMBER  NUMBER  NUMBER                           IN CONTROL
-------------------------------------------------------------------------------------------------------------------------
       1   00002     00002    20003    00002    20003    00002    20003   H11      MMM           SEQUENTIAL COORDINATION
       2   00002     00002    20003       -        -     00002    20003   H68      MXM           SEQUENTIAL COORDINATION
       3   00002     00002    20003       -        -        -        -       -     MXX           SEQUENTIAL COORDINATION
       4   00002     00002    20005    00002    20005       -        -       -     MMX           SEQUENTIAL COORDINATION
       5   00002     00002    20005       -        -        -        -       -     MXX           SEQUENTIAL COORDINATION
       6   00002        -        -       -        -        -        -       -     XXX      M    STANDARD PROCESSING
       7   00003     00003    30001    00003    30001    00003    30001   E68      MMM           SEQUENTIAL COORDINATION
       8   00003     00003    30001       -        -        -        -       -     MXX           SEQUENTIAL COORDINATION
       9   00003        -        -       -        -        -        -       -     XXX      M    STANDARD PROCESSING
      10   00005        -        -       -        -        -        -       -     XXX      X    STANDARD PROCESSING
      11   00010     00010    01001    00010    01001    00010    01001   N34      MMM           SEQUENTIAL COORDINATION
      12   00010     00010    01001       -        -        -        -       -     MXX           SEQUENTIAL COORDINATION
      13   00010     00010    01004    00010    01004    00010    01004   E68      MMM           SEQUENTIAL COORDINATION
      14   00010     00010    01004       -        -     00010    01004   M52      MXM           SEQUENTIAL COORDINATION
      15   00010     00010    01004       -        -        -        -       -     MXX           SEQUENTIAL COORDINATION
      16   00010        -        -       -        -        -        -       -     XXX      M    STANDARD PROCESSING
      17   00015        -        -       -        -        -        -       -     XXX      X    STANDARD PROCESSING
** MK4FT06  TYPE 1       67 FIELDS WERE PRINTED AS INVALID, MISSING OR OVERFLOW.                                  *****
** MK4FT03  TYPE 0  END OF REPORT.                                                                                 *****
** MK4W204  TYPE 0  NUMBER OF MESSAGES PRINTED IS    3.                                                            *****
                    M4REPO   -      29 RECORDS INPUT
                    M4LIST   -      30 RECORDS OUTPUT
** MK4W209  TYPE 0  1004272 BYTES OF MAIN STORAGE UNUSED DURING REPORTING PHASE                                    *****
```

Figure D-26 Chained Multi-File Sequential Coordination: The Application Coding and Report Part 2 (Page 3 of 3)

## User Coordination Examples Application – Banner Page

```
NOV 03, 2001  13.23.55                                                                                       PAGE      1
                                        ******************************************
                                        * VISION:Builder   4400  (OS/390 - 14.0)  *
                                        *           COPYRIGHT (C) 2002           *
                                        * COMPUTER ASSOCIATES INTERNATIONAL, INC. *
                                        ******************************************
VISION:Builder JCR1 14.0 110201              ENABLED FOR IBM LANGUAGE ENVIRONMENT                BUILD STAMP = 101306,12:51:24.
                                          COMLIB  MODIFICATIONS (PTFs, APARs) INSTALLED
                                    201,202,203,204,205,206,207,208,209,210,212,213,214,215,216
=============================================================================================================================
=                                                                                                                           =
=                    I N S T A L L A T I O N    P A R A M E T E R S    ( M 4 P A R A M S , M A R K L I B P )               =
=                                                                                                                           =
=  SYSTEM DELIMITER:   #          PAGE HEIGHT:       66         M4LIST WIDTH:      132        DEF WIDTH OF PAGE:  0        =
=                                                                                                                           =
=  AUTO GRAND:         N          HEADING CHAR:      -          SUBTITLE REPEAT:   N          INVALID FIELD:      *        =
=                                                                                                                           =
=  MISSING FIELD:      -          NON-EDIT FIELD:    +          PERCENT CHAR:      %          LEFT SEPARATOR:     (        =
=                                                                                                                           =
=  RIGHT SEPARATOR:    )          SINGLE SEPARATOR:  ,          SOURCE SPACING:    1          PRINT MESSAGES:     Y        =
=                                                                                                                           =
=  CONSOLE MESSAGES:   N          M4REPO BLOCKSIZE:  4,096      INPUT I/O BUFFERS: 2          OUTPUT I/O BUFFERS: 1        =
=                                                                                                                           =
=  SNGL-STEP STORAGE:  8,192      SNGL-STEP SORTSIZE: 524,288   DIGIT SELECT CHAR: 9          ZERO SUPPRESS CHAR: Z        =
=                                                                                                                           =
=  CURRENCY CHAR:      $          PLUS CHAR:         +          MINUS CHAR:        -          CHECK PROTECT CHAR: *        =
=                                                                                                                           =
=  DECIMAL CHAR:       .          GROUPING CHAR:     ,          PRIMARY PLOT CHAR: X          SECONDARY PLOT CHAR:*        =
```

Figure D-27 User Coordination Examples Application - Banner Page (Page 1 of 2)

```
=                                                                                                                                  =
=  FIT PLOT CHAR:      .          HORIZONTAL AXIS:     _          HORIZONTAL HASH:    |          VERTICAL AXIS:      |          =
=                                                                                                                                  =
=  VERTICAL HASH:      -          MINUTES/HOUR:        60         SECONDS/MINUTE:     60         TIME DELIMITER:     HH:MM:SS   =
=                                                                                                                                  =
=  DATE FORMAT:      MMM DD, YYYY  TODAY FORMAT+DELIM: MM/DD/YY   ISDATE DELIMITER:   YYYY-MM-DD  JULIAN DELIMITER:   YY.DDD     =
=                                                                                                                                  =
=  SORT PROGRAM CODE:  2          MINCORE VALUE:       12 K       ALT M4LIST WIDTH:   264        ALT DEF W/OF PAGE:  0          =
=                                                                                                                                  =
=  MAX LINES OF TRACE: 1,024      ITEM TRACKING:       0          SUPPRESS NDS REPT?: N          DEFAULT MAXGETMN:   1,024 K    =
=                                                                                                                                  =
=  CONDITION CODE 1:   0          CONDITION CODE 2:    4          CONDITION CODE 3:   0          CONDITION CODE 4:   16         =
=                                                                                                                                  =
==================================================================================================================================
```

Figure D-27 User Coordination Examples Application - Banner Page (Page 2 of 2)

## User Coordination Examples Application: Run Control – Part 1 of 7

```
NOV 03, 2001  13.23.55                                                                                                  PAGE      2
-----------------------------------------------------------------------------------------------------------------------------------
  RUN     STMT    FILE    O N S D U T R A B S D S L  RB  R S M R S            START           END             SEQUENCE   LIST
  NAME    TYPE    NAME    L E E I P R E U U R L C S  PL  J O O F S            SEARCH          SEARCH          73 TO 80    SEQ
                          D W Q R D N P D F T M N T  TK  T P P O R
-----------------------------------------------------------------------------------------------------------------------------------
(F160    ) (RC)  (MASTER1 ) (S)         (S)       (T,G)          (R)                                                (ASLGEN'D)     1

    CONTROL NAME F160 TERM LISTGEN
    FILE MASTER INPUT MASTER1
    FILE CORD1 CORDFILE USERREAD
    FILE CORD2 CORDFILE USERREAD
    FILE CORD3 CORDFILE GENERIC CUSTNUM
    FILE CORD4 CORDFILE GENERIC CUST04
    FILE CORD5 CORDFILE USERREAD
    FILE REPORT
-----------------------------------------------------------------------------------------------------------------------------------
  RUN     STMT  LOGICAL  LBL  DTF/DD      KEY       KEY       KEY      QUALIFIER   USER I/O  ARRAY      PASSWORD/  SEQUENCE   LIST
  NAME    TYPE  FILENAME       NAME     FIELD-1   FIELD-2   FIELD-3   ICF CRD DBI   MODULE   QUAL       AUTH.-ID   73 TO 80    SEQ
-----------------------------------------------------------------------------------------------------------------------------------
(F160    ) (RF)  (MASTER1 ) (S) (M4OLD   )                                                                          (ASLGEN'D)     2
(F160    ) (RF)  (CORDFILE) (S) (M4CORD1 )                                  (R)                                     (ASLGEN'D)     3
(F160    ) (RF)  (CORDFILE) (S) (M4CORD2 )                                  (R)                                     (ASLGEN'D)     4
(F160    ) (RF)  (CORDFILE) (S) (M4CORD3 )  (CUSTNUM )                      (G)                                     (ASLGEN'D)     5
(F160    ) (RF)  (CORDFILE) (S) (M4CORD4 )  (CUST04  )                      (G)                                     (ASLGEN'D)     6
(F160    ) (RF)  (CORDFILE) (S) (M4CORD5 )                                  (R)                                     (ASLGEN'D)     7
```

Figure D-28 User Coordination Examples Application: Run Control Part 1

## User Coordination Examples Application: Create Report of MASTER1 and CORDFILE Keys – Part 2 of 7

```
==================================================================================================================================
=                                                                                                                                  =
=                                                  F I L E   S U M M A R Y                                                         =
=                                                                                                                                  =
= JOBNAME: KREJO04X                          STEPNAME: F160                                                                        =
=                                                                                                                                  =
=  FILE NAME: M4OLD                          DSNAME: BUILDER.R140.SINGLE.MASTER.R06#1                                              =
=  FILE NAME: M4CORD1                        DSNAME: BUILDER.R140.CORD.FILE.R06#1                                                  =
=  FILE NAME: M4CORD2                        DSNAME: BUILDER.R140.CORD.FILE.R06#1                                                  =
=  FILE NAME: M4CORD3                        DSNAME: BUILDER.R140.CORD.FILE.R06#1                                                  =
=  FILE NAME: M4CORD4                        DSNAME: BUILDER.R140.CORD.FILE.R06#1                                                  =
=  FILE NAME: M4CORD5                        DSNAME: BUILDER.R140.CORD.FILE.R06#1                                                  =
=  FILE NAME: M4LIB                          DSNAME: BUILDER.R140.M4LIB.R06#1                                                      =
=                                                                                                                                  =
==================================================================================================================================
NOV 03, 2001  13.23.55                                                                                                  PAGE      3
                                                  ******************************
                                                  *  PROC NAME - MAIN          *
                                                  *  INPUT STREAM PROCEDURE    *
                                                  ******************************
-----------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                                 MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID           ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM       73 TO 80    SEQ
-----------------------------------------------------------------------------------------------------------------------------------
```

Figure D-29 User Coordination Examples Application: Create Report of MASTER1 and CORDFILE Keys Part 2 (Page 1 of 2)

```
(ER)                                                                    (N)                    (ASLGEN'D)    8
    MAIN: PROC                                                        00910000
    ;                                                                 00910000
    ; List the MASTER1 Record and the CORD FILE Records (A Serial Read)  00910000
    ;   Use SECTIONAL for Column Heading Lines                        00910000
    ;                                                                 00910000
    DO FORALL 5.ORDRSEG                                               00910000
       REPORT O.CUSTNUM 5.CUSTORDR                                    00910000
        GROUP BY O.CUSTNUM                                            00910000
        ITEM O.CUSTNUM  SPACES 03                                     00910000
        ITEM 5.CUSTORDR SPACES 10                                     00910000
        SECTION COLUMNHEADING                                         00910000
         LINE COL(01) '----- FILE KEYS VALUES -------'                00910000
         LINE COL(01) LIT('-',30)                                     00910000
         LINE COL(03) ' MASTER1     USER CORD FILES'                  00910000
         LINE COL(03) '  FILE       NUMBER 1,2,3,4'                   00910000
         LINE COL(01) LIT('-',30)                                     00910000
         LINE COL(03) 'CUSTOMER       CUSTOMER AND'                   00910000
         LINE COL(03) ' NUMBER        ORDER NUMBER'                   00910000
         LINE COL(01) LIT('-',30)                                     00910000
        END SECTION                                                   00910000
       END REPORT                                                     00910000
    END DO                                                            00910000
```

Figure D-29 User Coordination Examples Application: Create Report of MASTER1 and CORDFILE Keys Part 2 (Page 2 of 2)

## User Coordination Examples Application: Coding for Serial Read of CORD FILE 1 – Part 3 of 7

```
    ;                                                                 00910000
    ; Read CORD FILE 1 "SERIALLY" and Report ALL Records              00910000
    ;                                                                 00910000
    DO FORALL 1.ORDRSEG                                               00910000
       REPORT O.CUSTNUM 1.CUSTORDR F.RSTATUS                          00910000
        TITLE 'SERIALLY READ CORD FILE'                               00910000
        TITLE 'LIST FILE KEYS AND READ STATUS FLAG'                   00910000
       END REPORT                                                     00910000
    END DO                                                            00910000
```

Figure D-30 User Coordination Examples Application: Coding for Serial Read of CORD FILE 1 Part 3

## User Coordination Examples Application: Coding for DIRECT Processing of CORD FILE 2 – Part 4 of 7

```
    ;                                                                 00910000
    ; Read CORD FILE 2 "DIRECTLY" and Report "EQUAL" Record           00910000
    ;                                                                 00910000
    IF FIND(2.ORDRSEG WHERE 2.CUSTORDR = '0000220005')                00910000
       REPORT O.CUSTNUM 2.CUSTORDR F.RSTATUS                          00910000
        TITLE 'DIRECTLY READ CORD FILE'                               00910000
        TITLE 'LIST FILE KEYS AND READ STATUS FLAG'                   00910000
       END REPORT                                                     00910000
    END IF                                                            00910000
```

Figure D-31 User Coordination Examples Application: Coding for DIRECT Processing of CORD FILE 2 Part 4

## User Coordination Examples Application: Coding for Generic Key Processing of CORD FILE 3 – Part 5 of 7

```
    ;                                                                 00910000
    ; Read CORD FILE 3 "GENERICALLY' and Report Records within the    00910000
    ; CUSTNUM Group; having the same customer number                  00910000
    ;                                                                 00910000
    IF FIND(3.ORDRSEG WHERE 3.CUSTORDR >= O.CUSTNUM)                  00910000
      DO WHILE F.RSTATUS = 'KEQL' OR F.RSTATUS = 'KGTR'
```

Figure D-32 User Coordination Examples Application: Coding for Generic Key Processing of CORD FILE 3 Part 5 (Page 1 of 2)

```
        REPORT O.CUSTNUM 3.CUSTORDR F.RSTATUS                           00910000
         TITLE 'GENERIC READ CORD FILE BY CUSTOMER NUMBER'              00910000
         TITLE 'LIST FILE KEYS AND READ STATUS FLAG'                    00910000
NOV 03, 2001  13.23.55                                                                PAGE      4
        END REPORT                                                      00910000
        IF FIND(3.ORDRSEG)                                              00910000
        END IF                                                          00910000
      END DO                                                            00910000
    END IF                                                              00910000
```

Figure D-32 User Coordination Examples Application: Coding for Generic Key Processing of CORD FILE 3 Part 5 (Page 2 of 2)

## User Coordination Examples Application: Coding for Processing of CORD FILE 4 – Part 6 of 7

```
;                                                                       00910000
; Read CORD FILE 4 "GENERICALLY' and Report Records where the first     00910000
; four positions in the CUSTNUM match the same in the CORD FILE        00910000
;                                                                       00910000
IF FIND(4.ORDRSEG WHERE 4.CUSTORDR >= PF(N.CUSTNUM,1,4))                00910000
  DO WHILE F.RSTATUS = 'KEQL' OR F.RSTATUS = 'KGTR'                     00910000
    REPORT O.CUSTNUM 4.CUSTORDR F.RSTATUS                               00910000
     TITLE 'GENERIC READ CORD FILE BY CUSTOMER NUMBER'                  00910000
     TITLE 'LIST FILE KEYS AND READ STATUS FLAG'                        00910000
    END REPORT                                                          00910000
    IF FIND(4.ORDRSEG)                                                  00910000
    END IF                                                              00910000
  END DO                                                                00910000
END IF                                                                  00910000
;                                                                       00910000
END PROC                                                                00910000
```

Figure D-33 User Coordination Examples Application: Coding for Processing of CORD FILE 4 Part 6

## User Coordination Examples Application: Fixed Format Source Listing and Reports – Part 7 of 7

```
-------------------------------------------------------------------------------------------------------------------------------
STMT SEQ  LOG CON .....OPERAND-A.....  OPER  .............OPERAND-B.............   ......RESULT......  PARTIAL FIELD  SEQUENCE   LIST
TYPE NO.  LEV CTR QLF FIELD   SEG-LVL  ATN   QLF FIELD   SEG-LVL  (OR CONSTANT)   QLF FIELD  SEG-LVL  STRT NUMB OPR  73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(PR) (100)                            (GO)      (SUB LOOP___1                 )                                    (ASLGEN'D)     9
(PR) (101)                            (GO)      (SUB LOOP___3                 )                                    (ASLGEN'D)    10
(PR) (102)                            (R )    (C,0000220005                   ) (T,TEMP___5)                       (ASLGEN'D)    11
** MK4DI02  TYPE 0  DEFINITION OF TEMPORARY FIELD TEMP___5 - FORMAT C  LENGTH  10  DECIMALS 0 EDIT LENGTH  10 .                 *****
(PR) (103)         (T,TEMP___5)       (RE)    (2)                                                                  (ASLGEN'D)    12
(PR) (104)                            (NS)      (106                          )                                    (ASLGEN'D)    13
(PR) (105)                            (GO)      (SUB REPT___6                 )                                    (ASLGEN'D)    14
(PR) (106)         (O,CUSTNUM ) 001-1 (RG)    (3)                                                                  (ASLGEN'D)    15
(PR) (107)                            (NS)      (109                          )                                    (ASLGEN'D)    16
(PR) (108)                            (GO)      (SUB LOOP___7                 )                                    (ASLGEN'D)    17
(PR) (109)                            (R )    (N,CUSTNUM ) 001-1                (T,TEMP___9)        ( 1) ( 4) (B) (ASLGEN'D)    18
** MK4DI02  TYPE 0  DEFINITION OF TEMPORARY FIELD TEMP___9 - FORMAT C  LENGTH   4  DECIMALS 0 EDIT LENGTH   4 .                 *****
(PR) (110)         (T,TEMP___9)       (RG)    (4)                                                                  (ASLGEN'D)    19
(PR) (111)                            (NS)      (END                          )                                    (ASLGEN'D)    20
(PR) (112)                            (GO)      (SUB LOOP_10                  )                                    (ASLGEN'D)    21
** MK4CQ06  TYPE 0  O AND N FIELD QUALIFIERS UTILIZED IN THIS REQUEST.                                                  *****
NOV 03, 2001  13.23.55                                                                                      PAGE      5

                                        ****************************
                                        *  REQUEST NAME - LOOP___1  *
                                        *  GENERATED REQUEST        *
                                        ****************************

-------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                              MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID        ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM       73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(ER)                                                                          (S)          (99999)                 (ASLGEN'D)    22
-------------------------------------------------------------------------------------------------------------------------------
STMT SEQ  LOG CON .....OPERAND-A.....  OPER  .............OPERAND-B.............   ......RESULT......  PARTIAL FIELD  SEQUENCE   LIST
TYPE NO.  LEV CTR QLF FIELD   SEG-LVL  ATN   QLF FIELD   SEG-LVL  (OR CONSTANT)   QLF FIELD  SEG-LVL  STRT NUMB OPR  73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(PR) (100)                            (RD)    (5)                                                                  (ASLGEN'D)    23
(PR) (101)         (F,EOF     )       (EQ)    (C,Y                          )                    ( 7) ( 1) (A) (ASLGEN'D)    24
(PR) (102)                            (NS)      (RETURN                       )                                    (ASLGEN'D)    25
(PR) (103)                            (GO)      (SUB REPT___2                 )                                    (ASLGEN'D)    26
```

Figure D-34 User Coordination Examples Application: Fixed Format Source Listing and Reports Part 7 (Page 1 of 5)

```
(PR) (104)                         (GO)     (REQUEST LOOP___1              )                           (ASLGEN'D)    27
NOV 03, 2001  13.23.55                                                                                       PAGE     6
                                                ****************************
                                                *  REQUEST NAME - REPT___2  *
                                                *  GENERATED REQUEST        *
                                                ****************************
-----------------------------------------------------------------------------------------------------------------------
STMT REPORT                              MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ         BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID        ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM       73 TO 80    SEQ
-----------------------------------------------------------------------------------------------------------------------
(ER)                                                                           (S)                               (ASLGEN'D)    28
-----------------------------------------------------------------------------------------------------------------------
 ST SUM V  8  PG  PG   I SP  MAXIMUM TB COLUMN  DT  PG  STRT LIN SL RH .......SUBFILE........ C E A S R S N       C  SEQUENCE   LIST
 TY RPT S LPI WID HGT  M FRM LPP PGS TO HEADING POS POS PAGE NUM UB PN ENT    NAME   BLK/ FMT H F G T S U D       O  73 TO 80    SEQ
 MP     P              G             LT TYP POS          NUM      ML TD REC          BUF      C C S L T M S       N
-----------------------------------------------------------------------------------------------------------------------
(E1)                                                                                                             (ASLGEN'D)    29
-----------------------------------------------------------------------------------------------------------------------
         SP  Q              E  N  S  D  C  S  T  C  C  M  M  A  G  D Q PCT/RAT     P                 PARTIAL  SEQUENCE   LIST
STMT SEQ  BF  L  FIELD   SEG-  N  P  E  E  T  U  O  U  N  A  I  V  M  E L  FIELD   RA- C      PICTURE   STRT NO  73 TO 80    SEQ
TYPE NO.  COL F  NAME      LVL  D  R  Q  S  L  B  T  M  T  X  N  G  R  C F  NAME    TIO T       EDIT     CHR CHR
-----------------------------------------------------------------------------------------------------------------------
(R1)      ( 3) (O,CUSTNUM ) 001-1            (1)                                                          (ASLGEN'D)    30
(R1)      (10) (5,CUSTORDR) 001-1                                                                         (ASLGEN'D)    31
-----------------------------------------------------------------------------------------------------------------------
STMT  SEQ                                                                                                 SEQUENCE   LIST
TYPE  NO.                                          TEXT                                                   73 TO 80    SEQ
-----------------------------------------------------------------------------------------------------------------------
(F1)                         (COLUMN-HEADING SECTION ##                                    )              (ASLGEN'D)    32
(F1)                         (01C #----- FILE KEYS VALUES -------# ##                      )              (ASLGEN'D)    33
(F1)                         (01C 30#-# ##                                                 )              (ASLGEN'D)    34
(F1)                         (03C # MASTER1     USER CORD FILES# ##                        )              (ASLGEN'D)    35
(F1)                         (03C #  FILE       NUMBER 1,2,3,4# ##                         )              (ASLGEN'D)    36
(F1)                         (01C 30#-# ##                                                 )              (ASLGEN'D)    37
(F1)                         (03C #CUSTOMER       CUSTOMER AND# ##                         )              (ASLGEN'D)    38
(F1)                         (03C # NUMBER        ORDER NUMBER# ##                         )              (ASLGEN'D)    39
(F1)                         (01C 30#-# ##                                                 )              (ASLGEN'D)    40
NOV 03, 2001  13.23.55                                                                                       PAGE     7
                                                ****************************
                                                *  REQUEST NAME - LOOP___3  *
                                                *  GENERATED REQUEST        *
                                                ****************************
-----------------------------------------------------------------------------------------------------------------------
STMT REPORT                              MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ         BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID        ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM       73 TO 80    SEQ
-----------------------------------------------------------------------------------------------------------------------
(ER)                                                                           (S)          (99999)              (ASLGEN'D)    41
-----------------------------------------------------------------------------------------------------------------------
STMT SEQ  LOG CON .....OPERAND-A.....  OPER  ...........OPERAND-B............  ......RESULT......  PARTIAL FIELD SEQUENCE   LIST
TYPE NO.  LEV CTR QLF FIELD   SEG-LVL  ATN   QLF FIELD  SEG-LVL  (OR CONSTANT)  QLF FIELD  SEG-LVL  STRT NUMB OPR 73 TO 80    SEQ
-----------------------------------------------------------------------------------------------------------------------
(PR) (100)                         (RD)   (1)                                                             (ASLGEN'D)    42
(PR) (101)          (F,EOF     )   (EQ)   (C,Y                              )                ( 3)  ( 1)  (A) (ASLGEN'D)    43
(PR) (102)                         (NS)     (RETURN                         )                             (ASLGEN'D)    44
(PR) (103)                         (GO)     (SUB REPT___4                   )                             (ASLGEN'D)    45
(PR) (104)                         (GO)     (REQUEST LOOP___3               )                             (ASLGEN'D)    46
NOV 03, 2001  13.23.55                                                                                       PAGE     8
                                                ****************************
                                                *  REQUEST NAME - REPT___4  *
                                                *  GENERATED REQUEST        *
                                                ****************************
-----------------------------------------------------------------------------------------------------------------------
STMT REPORT                              MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ         BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID        ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM       73 TO 80    SEQ
-----------------------------------------------------------------------------------------------------------------------
(ER)                                                                           (S)                               (ASLGEN'D)    47
-----------------------------------------------------------------------------------------------------------------------
 ST SUM V  8  PG  PG   I SP  MAXIMUM TB COLUMN  DT  PG  STRT LIN SL RH .......SUBFILE........ C E A S R S N       C  SEQUENCE   LIST
 TY RPT S LPI WID HGT  M FRM LPP PGS TO HEADING POS POS PAGE NUM UB PN ENT    NAME   BLK/ FMT H F G T S U D       O  73 TO 80    SEQ
 MP     P              G             LT TYP POS          NUM      ML TD REC          BUF      C C S L T M S       N
-----------------------------------------------------------------------------------------------------------------------
(E1)                                                                                                             (ASLGEN'D)    48
-----------------------------------------------------------------------------------------------------------------------
         SP  Q              E  N  S  D  C  S  T  C  C  M  M  A  G  D Q PCT/RAT     P                 PARTIAL  SEQUENCE   LIST
STMT SEQ  BF  L  FIELD   SEG-  N  P  E  E  T  U  O  U  N  A  I  V  M  E L  FIELD   RA- C      PICTURE   STRT NO  73 TO 80    SEQ
TYPE NO.  COL F  NAME      LVL  D  R  Q  S  L  B  T  M  T  X  N  G  R  C F  NAME    TIO T       EDIT     CHR CHR
-----------------------------------------------------------------------------------------------------------------------
(R1)           (O,CUSTNUM ) 001-1                                                                         (ASLGEN'D)    49
(R1)           (1,CUSTORDR) 001-1                                                                         (ASLGEN'D)    50
(R1)           (F,RSTATUS )                                                                               (ASLGEN'D)    51
-----------------------------------------------------------------------------------------------------------------------
STMT  SEQ                                                                                                 SEQUENCE   LIST
TYPE  NO.                                          TEXT                                                   73 TO 80    SEQ
-----------------------------------------------------------------------------------------------------------------------
(T1)  (101)                  (SERIALLY READ CORD FILE#                                     )              (ASLGEN'D)    52
```

Figure D-34 User Coordination Examples Application: Fixed Format Source Listing and Reports Part 7 (Page 2 of 5)

```
(T1) (102)                  (LIST FILE KEYS AND READ STATUS FLAG#                          )                       (ASLGEN'D)   53
NOV 03, 2001  13.23.55                                                                                                    PAGE    9
                                                       *****************************
                                                       *  REQUEST NAME - REPT___6  *
                                                       *  GENERATED REQUEST        *
                                                       *****************************
-------------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                                  MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ         BACK BR RNT AB        SEQUENCE  LIST
TYPE  DATE          REQUESTOR ID           ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM        73 TO 80   SEQ
-------------------------------------------------------------------------------------------------------------------------------------
(ER)                                                                               (S)                               (ASLGEN'D)   54
-------------------------------------------------------------------------------------------------------------------------------------
 ST SUM V  8  PG  PG   I SP  MAXIMUM TB COLUMN  DT  PG  STRT LIN SL RH .......SUBFILE........ C E A S R S N       C  SEQUENCE  LIST
 TY RPT S LPI WID HGT  M FRM LPP PGS TO HEADING POS POS PAGE NUM UB PN ENT     NAME   BLK/ FMT H F G T S U D       O  73 TO 80   SEQ
 MP     P              G            LT TYP POS          NUM      ML TD REC            BUF     C C S L T M S       N
-------------------------------------------------------------------------------------------------------------------------------------
(E1)                                                                                                                 (ASLGEN'D)   55
-------------------------------------------------------------------------------------------------------------------------------------
          SP  Q                 E  N  S  D  C  S  T  C  C  M  M  A  G  D Q PCT/RAT     P                  PARTIAL   SEQUENCE  LIST
STMT SEQ  BF  L  FIELD   SEG-   N  P  E  E  T  U  O  U  N  A  I  V  M  E L  FIELD   RA- C      PICTURE    STRT NO   73 TO 80   SEQ
TYPE NO.  COL F  NAME     LVL  D  R  Q  S  L  B  T  M  T  X  N  G  R  C F  NAME     TIO T       EDIT      CHR CHR
-------------------------------------------------------------------------------------------------------------------------------------
(R1)          (O,CUSTNUM ) 001-1                                                                             (ASLGEN'D)   56
(R1)          (2,CUSTORDR) 001-1                                                                             (ASLGEN'D)   57
(R1)          (F,RSTATUS )                                                                                   (ASLGEN'D)   58
-------------------------------------------------------------------------------------------------------------------------------------
STMT  SEQ                                                                                                     SEQUENCE  LIST
TYPE  NO.                                                   TEXT                                              73 TO 80   SEQ
-------------------------------------------------------------------------------------------------------------------------------------
(T1)  (101)                 (DIRECTLY READ CORD FILE#                                      )                 (ASLGEN'D)   59
(T1)  (102)                 (LIST FILE KEYS AND READ STATUS FLAG#                          )                 (ASLGEN'D)   60
NOV 03, 2001  13.23.55                                                                                           PAGE   10
                                                       *****************************
                                                       *  REQUEST NAME - LOOP___7  *
                                                       *  GENERATED REQUEST        *
                                                       *****************************
-------------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                                  MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ         BACK BR RNT AB        SEQUENCE  LIST
TYPE  DATE          REQUESTOR ID           ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM        73 TO 80   SEQ
-------------------------------------------------------------------------------------------------------------------------------------
(ER)                                                                               (S)          (99999)              (ASLGEN'D)   61
-------------------------------------------------------------------------------------------------------------------------------------
STMT SEQ  LOG CON .....OPERAND-A.....  OPER  ...........OPERAND-B............  ......RESULT......  PARTIAL FIELD  SEQUENCE  LIST
TYPE NO.  LEV CTR QLF FIELD   SEG-LVL  ATN   QLF FIELD   SEG-LVL  (OR CONSTANT)  QLF FIELD  SEG-LVL  STRT NUMB OPR  73 TO 80   SEQ
-------------------------------------------------------------------------------------------------------------------------------------
(PR) (100)        (F,RSTATUS )         (EQ)   (C,KEQL                            )                             (ASLGEN'D)   62
(PR) (101)                             (GS)     (104                             )                             (ASLGEN'D)   63
(PR) (102)        (F,RSTATUS )         (EQ)   (C,KGTR                            )                             (ASLGEN'D)   64
(PR) (103)                             (NS)     (RETURN                          )                             (ASLGEN'D)   65
(PR) (104)                             (GO)     (SUB REPT___8                    )                             (ASLGEN'D)   66
(PR) (105)                             (RD)   (3)                                                              (ASLGEN'D)   67
(PR) (106)        (F,EOF     )         (EQ)   (C,Y                               )           ( 5)  ( 1)  (A)  (ASLGEN'D)   68
(PR) (107)                             (NS)     (108                             )                             (ASLGEN'D)   69
(PR) (108)                             (GO)     (REQUEST LOOP___7                )                             (ASLGEN'D)   70
NOV 03, 2001  13.23.55                                                                                           PAGE   11
                                                       *****************************
                                                       *  REQUEST NAME - REPT___8  *
                                                       *  GENERATED REQUEST        *
                                                       *****************************
-------------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                                  MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ         BACK BR RNT AB        SEQUENCE  LIST
TYPE  DATE          REQUESTOR ID           ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM        73 TO 80   SEQ
-------------------------------------------------------------------------------------------------------------------------------------
(ER)                                                                               (S)                               (ASLGEN'D)   71
-------------------------------------------------------------------------------------------------------------------------------------
 ST SUM V  8  PG  PG   I SP  MAXIMUM TB COLUMN  DT  PG  STRT LIN SL RH .......SUBFILE........ C E A S R S N       C  SEQUENCE  LIST
 TY RPT S LPI WID HGT  M FRM LPP PGS TO HEADING POS POS PAGE NUM UB PN ENT     NAME   BLK/ FMT H F G T S U D       O  73 TO 80   SEQ
 MP     P              G            LT TYP POS          NUM      ML TD REC            BUF     C C S L T M S       N
-------------------------------------------------------------------------------------------------------------------------------------
(E1)                                                                                                                 (ASLGEN'D)   72
-------------------------------------------------------------------------------------------------------------------------------------
          SP  Q                 E  N  S  D  C  S  T  C  C  M  M  A  G  D Q PCT/RAT     P                  PARTIAL   SEQUENCE  LIST
STMT SEQ  BF  L  FIELD   SEG-   N  P  E  E  T  U  O  U  N  A  I  V  M  E L  FIELD   RA- C      PICTURE    STRT NO   73 TO 80   SEQ
TYPE NO.  COL F  NAME     LVL  D  R  Q  S  L  B  T  M  T  X  N  G  R  C F  NAME     TIO T       EDIT      CHR CHR
-------------------------------------------------------------------------------------------------------------------------------------
(R1)          (O,CUSTNUM ) 001-1                                                                             (ASLGEN'D)   73
(R1)          (3,CUSTORDR) 001-1                                                                             (ASLGEN'D)   74
(R1)          (F,RSTATUS )                                                                                   (ASLGEN'D)   75
-------------------------------------------------------------------------------------------------------------------------------------
STMT  SEQ                                                                                                     SEQUENCE  LIST
TYPE  NO.                                                   TEXT                                              73 TO 80   SEQ
-------------------------------------------------------------------------------------------------------------------------------------
(T1)  (101)                 (GENERIC READ CORD FILE BY CUSTOMER NUMBER#                    )                 (ASLGEN'D)   76
(T1)  (102)                 (LIST FILE KEYS AND READ STATUS FLAG#                          )                 (ASLGEN'D)   77
NOV 03, 2001  13.23.55                                                                                           PAGE   12
```

Figure D-34 User Coordination Examples Application: Fixed Format Source Listing and Reports Part 7 (Page 3 of 5)

```
                                                ****************************
                                                *  REQUEST NAME - LOOP__10 *
                                                *  GENERATED REQUEST       *
                                                ****************************

-------------------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                                    MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID              ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM      73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------------------
(ER)                                                                                  (S)          (99999)             (ASLGEN'D)    78
-------------------------------------------------------------------------------------------------------------------------------------------
STMT SEQ  LOG CON .....OPERAND-A.....  OPER  .............OPERAND-B.............  ......RESULT......  PARTIAL FIELD  SEQUENCE   LIST
TYPE NO.  LEV CTR QLF FIELD   SEG-LVL  ATN   QLF FIELD   SEG-LVL  (OR CONSTANT)  QLF FIELD  SEG-LVL  STRT NUMB OPR  73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------------------
(PR) (100)         (F,RSTATUS )        (EQ)   (C,KEQL                       )                                      (ASLGEN'D)    79
(PR) (101)                             (GS)     (104                        )                                      (ASLGEN'D)    80
(PR) (102)         (F,RSTATUS )        (EQ)   (C,KGTR                       )                                      (ASLGEN'D)    81
(PR) (103)                             (NS)     (RETURN                     )                                      (ASLGEN'D)    82
(PR) (104)                             (GO)     (SUB REPT__11               )                                      (ASLGEN'D)    83
(PR) (105)                             (RD)   (4)                                                                   (ASLGEN'D)    84
(PR) (106)         (F,EOF     )        (EQ)   (C,Y                          )                  ( 6) ( 1) (A)      (ASLGEN'D)    85
(PR) (107)                             (NS)     (108                        )                                      (ASLGEN'D)    86
(PR) (108)                             (GO)     (REQUEST LOOP__10           )                                      (ASLGEN'D)    87
NOV 03, 2001  13.23.55                                                                                                    PAGE    13

                                                ****************************
                                                *  REQUEST NAME - REPT__11 *
                                                *  GENERATED REQUEST       *
                                                ****************************

-------------------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                                    MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID              ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM      73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------------------
(ER)                                                                                  (S)                              (ASLGEN'D)    88
-------------------------------------------------------------------------------------------------------------------------------------------
 ST SUM V  8  PG  PG   I SP  MAXIMUM TB COLUMN  DT  PG  STRT LIN SL RH .......SUBFILE........ C E A S R S N      C  SEQUENCE   LIST
 TY RPT S LPI WID HGT  M FRM LPP PGS TO HEADING POS POS PAGE NUM UB PN ENT    NAME   BLK/ FMT H F G T S U D      O  73 TO 80    SEQ
 MP     P             G           LT TYP POS         NUM      ML TD REC          BUF      C C S L T M S      N
-------------------------------------------------------------------------------------------------------------------------------------------
(E1)                                                                                                                      (ASLGEN'D)    89
-------------------------------------------------------------------------------------------------------------------------------------------
         SP  Q              E  N  S  D  C  S  T  C  C  M  M  A  G  D Q PCT/RAT     P                PARTIAL   SEQUENCE   LIST
STMT SEQ BF  L  FIELD  SEG-  N  P  E  E  T  U  O  U  N  A  I  V  M  E L  FIELD   RA- C     PICTURE    STRT NO   73 TO 80    SEQ
TYPE NO. COL F  NAME     LVL D  R  Q  S  L  B  T  M  T  X  N  G  R  C F  NAME    TIO T      EDIT      CHR CHR
-------------------------------------------------------------------------------------------------------------------------------------------
(R1)         (O,CUSTNUM ) 001-1                                                                                           (ASLGEN'D)    90
(R1)         (4,CUSTORDR) 001-1                                                                                           (ASLGEN'D)    91
(R1)         (F,RSTATUS )                                                                                                 (ASLGEN'D)    92
-------------------------------------------------------------------------------------------------------------------------------------------
STMT  SEQ                                                                                                                 SEQUENCE   LIST
TYPE  NO.                                          TEXT                                                                    73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------------------
(T1) (101)                 (GENERIC READ CORD FILE BY CUSTOMER NUMBER#              )                                    (ASLGEN'D)    93
(T1) (102)                 (LIST FILE KEYS AND READ STATUS FLAG#                    )                                    (ASLGEN'D)    94
NOV 03, 2001  13.23.55                                                                                                    PAGE    14
** MK4BP01  TYPE 1  REQUEST LOOP__1 HAS BACK BRANCHING AND USER CORD FILE READ OPERATIONS, ENDLESS LOOPING IS POSSIBLE.       *****
** MK4BP01  TYPE 1  REQUEST LOOP__3 HAS BACK BRANCHING AND USER CORD FILE READ OPERATIONS, ENDLESS LOOPING IS POSSIBLE.       *****
** MK4BP01  TYPE 1  REQUEST LOOP__7 HAS BACK BRANCHING AND USER CORD FILE READ OPERATIONS, ENDLESS LOOPING IS POSSIBLE.       *****
** MK4BP01  TYPE 1  REQUEST LOOP__10 HAS BACK BRANCHING AND USER CORD FILE READ OPERATIONS, ENDLESS LOOPING IS POSSIBLE.      *****
NOV 03, 2001  13.23.55                                                                                                    PAGE    15
** MK4W204  TYPE 0  NUMBER OF MESSAGES PRINTED IS    7.                                                                      *****
                    M4OLD    -       1 RECORDS INPUT
                    M4CORD1  -       9 RECORDS INPUT
                    M4CORD2  -       1 RECORDS INPUT
                    M4CORD3  -       3 RECORDS INPUT
                    M4CORD4  -       7 RECORDS INPUT
                    M4CORD5  -       9 RECORDS INPUT
                    M4REPO   -      87 RECORDS OUTPUT
                    M4INPUT  -      78 RECORDS INPUT
                    M4LIST   -     441 RECORDS OUTPUT
                      15 TRACKS ASSIGNED TO M4LIB    --   13 TRACKS NOT FULL. LIBRARY DIRECTORY BLOCKING FACTOR IS  800.
** MK4W209  TYPE 0  1012512 BYTES OF MAIN STORAGE UNUSED DURING DECODING PHASE                                             *****
                    1013016 BYTES OF MAIN STORAGE UNUSED DURING COMPILATION PHASE
                     565152 BYTES OF MAIN STORAGE UNUSED DURING PROCESSING PHASE
** MK4ED02  TYPE 0  REQUEST-REPT__2  REPORT-1  REQUESTOR ID-                                                              *****
                                                                                                                        PAGE 1

----- FILE KEYS VALUES -------
-------------------------------
   MASTER1      USER CORD FILES
    FILE        NUMBER 1,2,3,4
-------------------------------
  CUSTOMER       CUSTOMER AND
   NUMBER        ORDER NUMBER
-------------------------------
   00002          0000110001
                  0000110004
                  0000110005
```

Figure D-34 User Coordination Examples Application: Fixed Format Source Listing and Reports Part 7 (Page 4 of 5)

```
                0000220003
                0000220005
                0000330001
                0001001001
                0001001004
                0002002003
** MK4FT03  TYPE 0  END OF REPORT.                                                                                  *****
** MK4ED02  TYPE 0  REQUEST-REPT___4  REPORT-1  REQUESTOR ID-                                                       *****
                                                     SERIALLY READ CORD FILE                                  PAGE 1
                                                LIST FILE KEYS AND READ STATUS FLAG
-----------------------------------------------------------------------------------------------------------------------------
  CUSTOMER   CUSTOMER &   RSTATUS
   NUMBER   ORDER NUMBER
-----------------------------------------------------------------------------------------------------------------------------
   00002     0000110001    KEQL
   00002     0000110004    KEQL
   00002     0000110005    KEQL
   00002     0000220003    KEQL
   00002     0000220005    KEQL
   00002     0000330001    KEQL
   00002     0001001001    KEQL
   00002     0001001004    KEQL
   00002     0002002003    KEQL
** MK4FT03  TYPE 0  END OF REPORT.                                                                                  *****
** MK4ED02  TYPE 0  REQUEST-REPT___6  REPORT-1  REQUESTOR ID-                                                       *****
                                                     DIRECTLY READ CORD FILE                                  PAGE 1
                                                LIST FILE KEYS AND READ STATUS FLAG
-----------------------------------------------------------------------------------------------------------------------------
  CUSTOMER   CUSTOMER &   RSTATUS
   NUMBER   ORDER NUMBER
-----------------------------------------------------------------------------------------------------------------------------
   00002     0000220005    KEQL
** MK4FT03  TYPE 0  END OF REPORT.                                                                                  *****
** MK4ED02  TYPE 0  REQUEST-REPT___8  REPORT-1  REQUESTOR ID-                                                       *****
                                            GENERIC READ CORD FILE BY CUSTOMER NUMBER                          PAGE 1
                                                LIST FILE KEYS AND READ STATUS FLAG
-----------------------------------------------------------------------------------------------------------------------------
  CUSTOMER   CUSTOMER &   RSTATUS
   NUMBER   ORDER NUMBER
-----------------------------------------------------------------------------------------------------------------------------
   00002     0000220003    KEQL
   00002     0000220005    KEQL
** MK4FT03  TYPE 0  END OF REPORT.                                                                                  *****
** MK4ED02  TYPE 0  REQUEST-REPT__11  REPORT-1  REQUESTOR ID-                                                       *****
                                            GENERIC READ CORD FILE BY CUSTOMER NUMBER                          PAGE 1
                                                LIST FILE KEYS AND READ STATUS FLAG
-----------------------------------------------------------------------------------------------------------------------------
  CUSTOMER   CUSTOMER &   RSTATUS
   NUMBER   ORDER NUMBER
-----------------------------------------------------------------------------------------------------------------------------
   00002     0000110001    KEQL
   00002     0000110004    KEQL
   00002     0000110005    KEQL
   00002     0000220003    KEQL
   00002     0000220005    KEQL
   00002     0000330001    KEQL
** MK4FT03  TYPE 0  END OF REPORT.                                                                                  *****
** MK4W204  TYPE 0  NUMBER OF MESSAGES PRINTED IS   10.                                                             *****
                   M4REPO   -       86 RECORDS INPUT
                   M4LIST   -       72 RECORDS OUTPUT
** MK4W209  TYPE 0  1001576 BYTES OF MAIN STORAGE UNUSED DURING REPORTING PHASE                                     *****
```

Figure D-34 User Coordination Examples Application: Fixed Format Source Listing and Reports Part 7 (Page 5 of 5)

# Output File Processing

## Run Control Command Group with Old Master (M4OLD) and New Master (M4NEW) Entries

```
    CONTROL                        SCANONLY
     FILE MASTER INPUT, NAME CUSTMAST
     FILE MASTER OUTPUT
     ; Old master in (M4OLD) and new master out (M4NEW)
     ;
     ; FILE statements showing cord and subfile specification entries
     ;
     FILE CORD1 NAME ITEMASTX   DIRECT BY O.ITEMORD
     ;   Index directly into the coordinated file
     ;   using field ITEMORD from the old master file buffer
     ;
     FILE SUBF1 NAME CUSTSUB
     FILE SUBF2 NAME ITEMSUB
     FILE SUBF3 NAME ITEMDB   ; ITEMDB is also the DBD name of this subfile
     ;
     FILE SUBF0 NAME DEFS4SUB, AUTODEF ; This file holds file definitions
                                       ; generated for subfiles
     ;
     FILE REPORT
     FILE REP2  ALTREP2
     ; Alternate report file(M4REP2) entry
     ;
```

```
NOV 06, 2001  16.51.41                                                                                    PAGE      3
                                                  ******************************
                                                  *  PROC NAME - MAIN          *
                                                  *  INPUT STREAM PROCEDURE    *
                                                  ******************************
     MAIN: PROC
     ; Each master file record control procedure
     LET T.TEMP1 = 'EMPLOYEE'
     LET T.COUNT = T.COUNT+1
     LET T.OTHER = 'OTHER'
     END PROC
     ;
** MK4DI02  TYPE 0  DEFINITION OF TEMPORARY FIELD TEMP1    - FORMAT C  LENGTH   8  DECIMALS 0 EDIT LENGTH   8 .                *****
** MK4DI02  TYPE 0  DEFINITION OF TEMPORARY FIELD COUNT    - FORMAT P  LENGTH   8  DECIMALS 0 EDIT LENGTH  20 .                *****
** MK4DI02  TYPE 0  DEFINITION OF TEMPORARY FIELD OTHER    - FORMAT C  LENGTH   5  DECIMALS 0 EDIT LENGTH   5 .                *****
NOV 06, 2001  16.51.41                                                                                    PAGE      4
                                                  ******************************
                                                  *  PROC NAME - REP1          *
                                                  *  INPUT STREAM PROCEDURE    *
                                                  ******************************
-----------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                                  MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE           REQUESTOR ID          ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM       73 TO 80    SEQ
-----------------------------------------------------------------------------------------------------------------------------------
(ER)                                                                                  (N)                            (ASLGEN'D)     1
     REP1: REPORT ITEMS T.TEMP1  FILE ALTREP2
     END REPORT
     ; Create report with output directed to the alternate report file
     ;
NOV 06, 2001  16.51.41                                                                                    PAGE      5
                                                  ******************************
                                                  *  PROC NAME - CUSTPART      *
                                                  *  INPUT STREAM PROCEDURE    *
                                                  ******************************
-----------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                                  MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE           REQUESTOR ID          ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM       73 TO 80    SEQ
-----------------------------------------------------------------------------------------------------------------------------------
(ER)                                                                                  (N)                            (ASLGEN'D)     2
     CUSTPART: EXTRACT   FILE CUSTSUB,   AUTODEF,
               RECFM VARIABLE,  BLKSIZE 1000 ,
         ITEMS O.CUSTNO, 1.ITEMNO, T.COUNT, F.DATE, T.OTHER, 1.ITEMNAME
     ; Create variable record format subfile and
     ; automatically generate the corresponding file definition
     ;
** MK4TY05  TYPE 0  SUBFILE RECORD LENGTH COMPUTED FROM THE ABOVE SPECIFICATION IS    61.                                  *****
NOV 06, 2001  16.51.41                                                                                    PAGE      6
                                                  ******************************
                                                  *  PROC NAME - ITEMSALL      *
                                                  *  INPUT STREAM PROCEDURE    *
                                                  ******************************
-----------------------------------------------------------------------------------------------------------------------------------
```

Figure D-35 Run Control Command Group with Old Master (M4OLD) and New Master (M4NEW) Entries (Page 1 of 2)

```
STMT REPORT                              MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID        ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM      73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(ER)                                                                          (N)                            (ASLGEN'D)     3
    ITEMSALL: EXTRACT   FILE ITEMSUB     ENTIRE 1
    ; Subfile extracting entire records from M4CORD1
NOV 06, 2001  16.51.41                                                                                          PAGE      7
                                          ****************************
                                          *  PROC NAME - ITEMSIMS    *
                                          *  INPUT STREAM PROCEDURE  *
                                          ****************************
-------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                              MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID        ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM      73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(ER)                                                                          (N)                            (ASLGEN'D)     4
    ITEMSIMS: EXTRACT   FILE ITEMDB      ENTIRE 0
    ; Subfile extracting entire records from old master buffer
    ;                                    into DLI database
NOV 06, 2001  16.51.41                                                                                          PAGE      8
** MK4PI05  TYPE 0  SCAN COMPLETE -- SCAN ONLY SPECIFIED.                                                                *****
NOV 06, 2001  16.51.41                                                                                          PAGE      9
** MK4W204  TYPE 0  NUMBER OF MESSAGES PRINTED IS    5.                                                                  *****
                   M4SUBF0  -       14 RECORDS OUTPUT
                   M4INPUT  -       44 RECORDS INPUT
                   M4LIST   -      161 RECORDS OUTPUT
                     15 TRACKS ASSIGNED TO M4LIB    --   12 TRACKS NOT FULL. LIBRARY DIRECTORY BLOCKING FACTOR IS  800.
** MK4W209  TYPE 0  1012584 BYTES OF MAIN STORAGE UNUSED DURING DECODING PHASE                                        *****
```

Figure D-35 Run Control Command Group with Old Master (M4OLD) and New Master (M4NEW) Entries (Page 2 of 2)

## DB2 Subfile Specification

```
-------------------------------------------------------------------------------------------------------------------------------
  RUN     STMT    FILE    O N S D U T R A B S D S L  RB  R S M R S          START          END             SEQUENCE   LIST
  NAME    TYPE    NAME    L E E I P R E U U R L C S  PL  J O O F S          SEARCH         SEARCH          73 TO 80    SEQ
                          D W Q R D N P D F T M N T  TK  T P P O R
-------------------------------------------------------------------------------------------------------------------------------
(ASLRC   ) (RC) (TEMPL   ) (S) (U)              (S,G)                                                          (ASLGEN'D)     1
    CONTROL DB2 D61A INM4CALL   LISTGEN     SCANONLY
    FILE MASTER INPUT, NAME TEMPL, KEYS NONE
    ;
    FILE SUBF3, NAME SQLOUT, TABLE "HANME02.TESTABLE" CREATE
    ;
-------------------------------------------------------------------------------------------------------------------------------
  RUN     STMT  LOGICAL  LBL  DTF/DD     KEY       KEY       KEY     QUALIFIER  USER I/O  ARRAY      PASSWORD/  SEQUENCE   LIST
  NAME    TYPE  FILENAME       NAME     FIELD-1   FIELD-2   FIELD-3  ICF CRD DBI  MODULE   QUAL       AUTH.-ID  73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(ASLRC   ) (RF) (TEMPL   ) (S) (M4OLD   )                                                                      (ASLGEN'D)     2
(ASLRC   ) (RF) (SQLOUT  ) (S) (M4SUBF3 )                                                                      (ASLGEN'D)     3
-------------------------------------------------------------------------------------------------------------------------------
  RUN     STMT    FILE     TABLE  AUTH.-ID /                  DATABASE /  TABLESPACE /                        SEQUENCE   LIST
  NAME    TYPE    NAME    CONTROL CREATOR-ID  TABLE NAME      OWNER-ID     DBSPACE                         73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(ASLRC   ) (RT) (M4SUBF3 )   (C)   (HANME02 ) (TESTABLE         )                                              (ASLGEN'D)     4
-------------------------------------------------------------------------------------------------------------------------------
  RUN     STMT  PARAMETER                                                                                      SEQUENCE   LIST
  NAME    TYPE    NAME                              PARAMETER VALUE                                            73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(ASLRC   ) (RP) (DB2     )       (D61A    INM4CALL                                 )                           (ASLGEN'D)     5


===============================================================================================================================
=                                                                                                                             =
=                                          F I L E   S U M M A R Y                                                            =
=                                                                                                                             =
= JOBNAME: HANME022                       STEPNAME: M4RUN                                                                     =
=                                                                                                                             =
=  FILE NAME: M4OLD                       TABLE:  DSN8610.EMP                                                                 =
=  FILE NAME: M4SUBF3                     TABLE:  HANME02.TESTABLE                                                            =
=  FILE NAME: M4LIB                       DSNAME: HANME02.COMLIB14                                                            =
=                                                                                                                             =
===============================================================================================================================
NOV 08, 2001  12.25.04                                                                                          PAGE      3
                                          ****************************
                                          *  PROC NAME - SUBFILE3    *
                                          *  INPUT STREAM PROCEDURE  *
                                          ****************************
-------------------------------------------------------------------------------------------------------------------------------
```

Figure D-36 DB2 Subfile Specification (Page 1 of 2)

```
STMT REPORT                              MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID         ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM      73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(ER)                                                                    (N)                                    (ASLGEN'D)     6
    SUBFILE3: EXTRACT FILE SQLOUT, RECFM DB2,
       ITEMS N.EMPNO N.JOB N.LASTNAME N.FIRSTNME
-------------------------------------------------------------------------------------------------------------------------------
 ST SUM V  8  PG  PG   I SP  MAXIMUM TB COLUMN  DT  PG  STRT LIN SL RH .......SUBFILE..... C E A S R S N      C  SEQUENCE   LIST
 TY RPT S LPI WID HGT  M FRM LPP PGS TO HEADING POS POS PAGE NUM UB PN ENT    NAME   BLK/ FMT H F G T S U D      O  73 TO 80    SEQ
 MP     P            G           LT TYP POS         NUM      ML TD REC          BUF      C C S L T M S      N
-------------------------------------------------------------------------------------------------------------------------------
(E1)                                                    (NR)   (SQLOUT  )      (S)                          (ASLGEN'D)     7
-------------------------------------------------------------------------------------------------------------------------------
        SP  Q              E N S D C S T C C M M A G D Q PCT/RAT     P                PARTIAL   SEQUENCE   LIST
STMT SEQ BF  L  FIELD  SEG- N P E E T U O U N A I V M E L FIELD   RA- C     PICTURE     STRT NO  73 TO 80    SEQ
TYPE NO. COL F  NAME     LVL D R Q S L B T M T X N G R C F NAME    TIO T      EDIT      CHR CHR
-------------------------------------------------------------------------------------------------------------------------------
(R1)         (N,EMPNO   ) 001-1                                                                (ASLGEN'D)     8
(R1)         (N,JOB     ) 001-1                                                                (ASLGEN'D)     9
(R1)         (N,LASTNAME) 001-1                                                                (ASLGEN'D)    10
(R1)         (N,FIRSTNME) 001-1                                                                (ASLGEN'D)    11
NOV 08, 2001  12.25.04                                                                             PAGE     4
** MK4PI05  TYPE 0  SCAN COMPLETE -- SCAN ONLY SPECIFIED.                                                    *****
NOV 08, 2001  12.25.04                                                                             PAGE     5
                                  ******************************************
                                  *   SUMMARY OF GENERATED SQL STATEMENTS   *
                                  ******************************************
-------------------------------------------------------------------------------------------------------------------------------
FILE>SEGMENT                  SQL STATEMENT                                          COMMENTS
-------------------------------------------------------------------------------------------------------------------------------
M4OLD>TEMPL       STATEMENT  1
          SELECT EMPNO,FIRSTNME,LASTNAME,JOB FROM DSN8610.EMP X1          STANDARD SELECT FOR SEQUENTIAL READ OF ROOT SEGMENT.
                                                                          TABLE ORDERED IN ASCENDING SEQUENCE. DB2 CALL(S): 1
                                                                          PREPARE(S),   OPEN(S), AND   FETCH(ES).
NOV 08, 2001  12.25.04                                                                             PAGE     6
** MK4W204  TYPE 0  NUMBER OF MESSAGES PRINTED IS    1.                                                      *****
                 M4INPUT  -       7 RECORDS INPUT
                 M4LIST   -     127 RECORDS OUTPUT
                   15 TRACKS ASSIGNED TO M4LIB    --   12 TRACKS NOT FULL. LIBRARY DIRECTORY BLOCKING FACTOR IS  800.
** MK4W209  TYPE 0  1016128 BYTES OF MAIN STORAGE UNUSED DURING DECODING PHASE                              *****
```

Figure D-36 DB2 Subfile Specification (Page 2 of 2)

## Report Summary File Specification

```
-------------------------------------------------------------------------------------------------------------------------------
   RUN     STMT    FILE    O N S D U T R A B S D S L  RB  R S M R S          START          END            SEQUENCE   LIST
   NAME    TYPE    NAME    L E E I P R E U U R L C S  PL  J O O F S          SEARCH         SEARCH          73 TO 80    SEQ
                           D W Q R D N P D F T M N T  TK  T P P O R
-------------------------------------------------------------------------------------------------------------------------------
(ASLRC   ) (RC) (TEMPL   ) (S) (U)     (S)         (G)          (R)                                        (ASLGEN'D)     1
    CONTROL DB2 D61A INM4CALL LISTGEN
    FILE MASTER INPUT, NAME TEMPL, KEYS NONE
    FILE REPORT
    FILE SUBF0 NAME DEFS AUTODEF ; holds generated file definitions
                                 ; for the report summary file
    ;
-------------------------------------------------------------------------------------------------------------------------------
   RUN     STMT  LOGICAL  LBL  DTF/DD      KEY       KEY       KEY      QUALIFIER   USER I/O  ARRAY       PASSWORD/  SEQUENCE   LIST
   NAME    TYPE  FILENAME       NAME      FIELD-1   FIELD-2   FIELD-3   ICF CRD DBI   MODULE    QUAL        AUTH.-ID  73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(ASLRC   ) (RF) (TEMPL   ) (S) (M4OLD   )                                                                   (ASLGEN'D)     2
(ASLRC   ) (RF) (DEFS    ) (S) (M4SUBF0 )                                  (C)                              (ASLGEN'D)     3
-------------------------------------------------------------------------------------------------------------------------------
   RUN     STMT  PARAMETER                                                                                   SEQUENCE   LIST
   NAME    TYPE    NAME                          PARAMETER VALUE                                           73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(ASLRC   ) (RP) (DB2     )       (D61A    INM4CALL                              )                          (ASLGEN'D)     4


=================================================================================================================
=                                                                                                               =
=                                          F I L E   S U M M A R Y                                              =
=                                                                                                               =
= JOBNAME: HANME022                        STEPNAME: M4RUN                                                      =
=                                                                                                               =
=  FILE NAME: M4OLD                        TABLE:  DSN8610.EMP                                                  =
=  FILE NAME: M4SUBF0                      DSNAME: HANME02.HANME022.JOB14428.D0000104.?                         =
=  FILE NAME: M4LIB                        DSNAME: HANME02.COMLIB14                                             =
=                                                                                                               =
=================================================================================================================
```

Figure D-37 Report Summary File Specification (Page 1 of 4)

```
NOV 08, 2001  12.40.18                                                                                                  PAGE      3
                                                    ****************************
                                                    *  PROC NAME - MAIN        *
                                                    *  INPUT STREAM PROCEDURE  *
                                                    ****************************

-------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                                MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID         ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM       73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(ER)                                                                              (N)                               (ASLGEN'D)      5
    MAIN: PROC
          LET T.COMP = SALARY + BONUS + COMM
          CALL REPORT COMPARE
          CALL REPORT DEPTSUM
    END PROC
    ;
-------------------------------------------------------------------------------------------------------------------------------
STMT   FIELD    FIELD FLD DEC OUTPT EDIT      INITIAL         LINE 1 OF       LINE 2 OF                                 SEQUENCE   LIST
TYPE   NAME     LNGTH TYP PLC EDIT  LGTH       VALUE        COLUMN HEADING  COLUMN HEADING                             73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(TF) (TEMP___3) (  6) (P) (2)                                                                                          (ASLGEN'D)      6
-------------------------------------------------------------------------------------------------------------------------------
STMT SEQ  LOG CON .....OPERAND-A.....  OPER  .............OPERAND-B.............   ......RESULT......  PARTIAL FIELD  SEQUENCE   LIST
TYPE NO.  LEV CTR QLF FIELD   SEG-LVL  ATN   QLF FIELD   SEG-LVL  (OR CONSTANT)   QLF FIELD  SEG-LVL  STRT NUMB OPR  73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(PR) (100)          (SALARY  ) 001-1  (+ )     (BONUS   ) 001-1                  (T,TEMP___3)                          (ASLGEN'D)      7
(PR) (101)        (T,TEMP___3)        (+ )     (COMM    ) 001-1                  (T,COMP___ )                          (ASLGEN'D)      8
** MK4DI02  TYPE 0  DEFINITION OF TEMPORARY FIELD COMP     - FORMAT P  LENGTH   6  DECIMALS 2 EDIT LENGTH  15 .                  *****
(PR) (102)                            (GO)     (SUB COMPARE                   )                                      (ASLGEN'D)      9
(PR) (103)                            (GO)     (SUB DEPTSUM                   )                                      (ASLGEN'D)     10
NOV 08, 2001  12.40.18                                                                                                  PAGE      4
                                                    ****************************
                                                    *  PROC NAME - COMPARE     *
                                                    *  INPUT STREAM PROCEDURE  *
                                                    ****************************

-------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                                MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID         ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM       73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(ER) (TODAY )                                                                     (S)                               (ASLGEN'D)     11
    COMPARE: REPORT SUMMARY ONLY, TYPE SUBROUTINE,
                ITEMS WORKDEPT, SEX, SALARY, BONUS, COMM
             FORMAT SUMFILE SUMRPTDD, AUTODEF, HEADINGS NAME
              ; report is captured in summary report file
              ;       identified with DD SUMRPTDD
              ; a file definition for that report file is generated
              ;       and captured in the definition subfile SUBF0
              ;       identified with DD M4SUBF0
              ;
              ORDER BY WORKDEPT, SEX
               GROUP AT LEVEL 2 BY WORKDEPT, SEX
                AVERAGE SALARY BY SEX
                AVERAGE BONUS  BY SEX
                AVERAGE COMM   BY SEX
                MAX SALARY BY SEX
                MAX BONUS  BY SEX
                MAX COMM   BY SEX
                COUNT BONUS BY WORKDEPT
           TITLE 'COMPENSATION COMPARISON BY DEPARTMENT AND GENDER'
    END REPORT
    ;
-------------------------------------------------------------------------------------------------------------------------------
 ST SUM V  8  PG  PG   I SP  MAXIMUM TB COLUMN  DT  PG  STRT LIN SL RH .......SUBFILE..... C E A S R S N      C  SEQUENCE   LIST
 TY RPT S LPI WID HGT  M FRM LPP PGS TO HEADING POS POS PAGE NUM UB PN ENT    NAME   BLK/ FMT H F G T S U D      O  73 TO 80    SEQ
 MP     P             G            LT TYP POS        NUM      ML TD REC           BUF      C C S L T M S      N
-------------------------------------------------------------------------------------------------------------------------------
(E1) (Y)                                (F)                      (SD)    (SUMRPTDD)                                (ASLGEN'D)     12
-------------------------------------------------------------------------------------------------------------------------------
         SP  Q                E N S  D  C  S  T  C  C  M  M  A  G  D Q PCT/RAT      P                PARTIAL   SEQUENCE   LIST
STMT SEQ BF  L  FIELD   SEG-   N P E  E  T  U  O  U  N  A  I  V  M  E L  FIELD   RA- C     PICTURE     STRT NO   73 TO 80    SEQ
TYPE NO. COL F  NAME      LVL  D R Q  S  L  B  T  M  T  X  N  G  R  C F  NAME    TIO T      EDIT       CHR CHR
-------------------------------------------------------------------------------------------------------------------------------
(R1)           (WORKDEPT) 001-1      (1)   (2)                                                                      (ASLGEN'D)     13
(R1)           (SEX     ) 001-1      (2)   (2)                                                                      (ASLGEN'D)     14
(R1)           (SALARY  ) 001-1                          (2)   (2)                                                  (ASLGEN'D)     15
(R1)           (BONUS   ) 001-1                        (2) (2)   (2)                                                (ASLGEN'D)     16
(R1)           (COMM    ) 001-1                          (2)   (2)                                                  (ASLGEN'D)     17
-------------------------------------------------------------------------------------------------------------------------------
STMT  SEQ                                                                                                          SEQUENCE   LIST
TYPE  NO.                                          TEXT                                                            73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(T1) (101)                 (COMPENSATION COMPARISON BY DEPARTMENT AND GENDER#         )                            (ASLGEN'D)     18
** MK4TY14  TYPE 0  THE RECORD LENGTH OF THE REPORT SUMMARY FILE FOR THIS REPORT IS    97.                                 *****
NOV 08, 2001  12.40.18                                                                                                  PAGE      5
                                                    ****************************
                                                    *  PROC NAME - DEPTSUM     *
```

Figure D-37 Report Summary File Specification (Page 2 of 4)

```
                                        *  INPUT STREAM PROCEDURE   *
                                        ****************************
-------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                              MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB      SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID        ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM      73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(ER)(TODAY )                                                                    (S)                            (ASLGEN'D)     19
    DEPTSUM: REPORT SUMMARY ONLY, TYPE SUBROUTINE,
             ITEMS WORKDEPT, SALARY, BONUS, COMM, T.COMP
        FORMAT HEADINGS NAME, METHOD ALTLIST DDNAME DEPTTOTS
         ; report is captured in alternate list file
         ;        identified with DD DEPTTOTS
         ; if DDNAME is not specified, default is M4LIST1
         ;
         ORDER BY WORKDEPT
          GROUP BY WORKDEPT
           TOTAL SALARY BY WORKDEPT
           TOTAL BONUS  BY WORKDEPT
           TOTAL COMM   BY WORKDEPT
           TOTAL T.COMP BY WORKDEPT
       TITLE 'COMPENSATION TOTALS BY DEPARTMENT'
    END REPORT
-------------------------------------------------------------------------------------------------------------------------------
 ST SUM V  8  PG  PG   I SP  MAXIMUM TB COLUMN  DT  PG  STRT LIN SL RH .......SUBFILE..... C E A S R S N      C  SEQUENCE   LIST
 TY RPT S LPI WID HGT  M FRM LPP PGS TO HEADING POS POS PAGE NUM UB PN ENT    NAME   BLK/ FMT H F G T S U D      O  73 TO 80    SEQ
 MP     P             G            LT TYP POS        NUM      ML TD REC          BUF      C C S L T M S      N
-------------------------------------------------------------------------------------------------------------------------------
(E1)(Y)                                 (F)                     (AL)   (DEPTTOTS)                             (ASLGEN'D)     20
-------------------------------------------------------------------------------------------------------------------------------
         SP  Q              E N S  D  C  S  T  C  C  M  M  A  G  D Q PCT/RAT     P                PARTIAL   SEQUENCE   LIST
STMT SEQ BF  L  FIELD   SEG- N P E  E  T  U  O  U  N  A  I  V  G  M  E L  FIELD   RA- C     PICTURE     STRT NO  73 TO 80    SEQ
TYPE NO. COL F  NAME     LVL D R Q  S  L  B  T  M  T  X  N  G  R  C F  NAME    TIO T      EDIT      CHR CHR
-------------------------------------------------------------------------------------------------------------------------------
(R1)          (WORKDEPT) 001-1      (1)   (1)                                                        (ASLGEN'D)     21
(R1)          (SALARY  ) 001-1                  (1)                                                 (ASLGEN'D)     22
(R1)          (BONUS   ) 001-1                  (1)                                                 (ASLGEN'D)     23
(R1)          (COMM    ) 001-1                  (1)                                                 (ASLGEN'D)     24
(R1)        (T,COMP    )                        (1)                                                 (ASLGEN'D)     25
-------------------------------------------------------------------------------------------------------------------------------
STMT  SEQ                                                                                                SEQUENCE   LIST
TYPE  NO.                                        TEXT                                                    73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(T1) (101)                (COMPENSATION TOTALS BY DEPARTMENT#                      )                     (ASLGEN'D)     26
NOV 08, 2001  12.40.18                                                                                       PAGE      6
                                        ******************************************
                                        *   SUMMARY OF GENERATED SQL STATEMENTS   *
                                        ******************************************
-------------------------------------------------------------------------------------------------------------------------------
FILE>SEGMENT                  SQL STATEMENT                                                  COMMENTS
-------------------------------------------------------------------------------------------------------------------------------
M4OLD>TEMPL       STATEMENT  1
          SELECT EMPNO,WORKDEPT,SEX,SALARY,BONUS,COMM FROM          STANDARD SELECT FOR SEQUENTIAL READ OF ROOT SEGMENT.
          DSN8610.EMP X1                                            TABLE ORDERED IN ASCENDING SEQUENCE. DB2 CALL(S): 1
                                                                    PREPARE(S), 1 OPEN(S), AND 49 FETCH(ES).
NOV 08, 2001  12.40.18                                                                                       PAGE      7
** MK4W204  TYPE 0  NUMBER OF MESSAGES PRINTED IS    2.                                                          *****
                   M4OLD    -      49 RECORDS INPUT
                   M4REPO   -     134 RECORDS OUTPUT
                   M4SUBF0  -      77 RECORDS OUTPUT
                   M4INPUT  -      48 RECORDS INPUT
                   M4LIST   -     226 RECORDS OUTPUT
                     15 TRACKS ASSIGNED TO M4LIB    --   12 TRACKS NOT FULL. LIBRARY DIRECTORY BLOCKING FACTOR IS  800.
** MK4W209  TYPE 0  1015520 BYTES OF MAIN STORAGE UNUSED DURING DECODING PHASE                                  *****
                   1016816 BYTES OF MAIN STORAGE UNUSED DURING COMPILATION PHASE
                   1060864 BYTES OF MAIN STORAGE UNUSED DURING PROCESSING PHASE
** MK4ED02  TYPE 0  REQUEST-COMPARE   REPORT-1  REQUESTOR ID-                                                  *****
11/08/01                              COMPENSATION COMPARISON BY DEPARTMENT AND GENDER                      PAGE 1
-------------------------------------------------------------------------------------------------------------------------------
              WORKDEPT  SEX     SALARY         BONUS         COMM
-------------------------------------------------------------------------------------------------------------------------------
WORKDEPT COUNT    A00      F                        5
        MAX.                        52,750.00      1,000.00      4,220.00
        AVG.                        51,500.00      1,000.00      4,220.00
WORKDEPT COUNT    A00      M                        3
        MAX.                        46,500.00        900.00      3,720.00
        AVG.                        35,000.00        700.00      2,800.00
WORKDEPT COUNT    B01      M                        1
        MAX.                        41,250.00        800.00      3,300.00
        AVG.                        41,250.00        800.00      3,300.00
WORKDEPT COUNT    C01      F                        7
        MAX.                        38,250.00        800.00      3,060.00
        AVG.                        31,972.85        671.42      2,558.00
WORKDEPT COUNT    D11      F                        4
        MAX.                        29,840.00        600.00      2,387.00
        AVG.                        25,817.50        525.00      2,065.25
WORKDEPT COUNT    D11      M                        7
```

Figure D-37 Report Summary File Specification (Page 3 of 4)

```
        MAX.                    32,250.00        600.00       2,580.00
        AVG.                    24,764.28        500.00       1,980.71
WORKDEPT COUNT    D21      F                          3
        MAX.                    36,170.00        700.00       2,893.00
        AVG.                    26,933.33        500.00       2,154.33
WORKDEPT COUNT    D21      M                          4
        MAX.                    28,760.00        600.00       2,301.00
        AVG.                    24,720.00        500.00       1,977.50
WORKDEPT COUNT    E01      M                          1
        MAX.                    40,175.00        800.00       3,214.00
        AVG.                    40,175.00        800.00       3,214.00
WORKDEPT COUNT    E11      F                          6
        MAX.                    29,750.00        600.00       2,380.00
        AVG.                    23,383.33        450.00       1,870.66
WORKDEPT COUNT    E11      M                          2
        MAX.                    17,750.00        400.00       1,420.00
        AVG.                    16,545.00        350.00       1,323.50
WORKDEPT COUNT    E21      F                          1
        MAX.                    25,370.00        500.00       2,030.00
        AVG.                    25,370.00        500.00       2,030.00
WORKDEPT COUNT    E21      M                          5
        MAX.                    26,150.00        500.00       2,092.00
        AVG.                    23,830.00        480.00       1,906.40
** MK4FT03  TYPE 0  END OF REPORT.                                                                  *****
** MK4DC15  TYPE 0  FOR THIS REPORT THE ALTERNATE M4LIST FILE IS DEPTTOTS.                          *****
** MK4ED02  TYPE 0  REQUEST-DEPTSUM   REPORT-1  REQUESTOR ID-                                       *****
** MK4FT09  TYPE 0  THIS REPORT HAD A TOTAL OF       12 LINES OF OUTPUT.                            *****
** MK4FT03  TYPE 0  END OF REPORT.                                                                  *****
** MK4W204  TYPE 0  NUMBER OF MESSAGES PRINTED IS    6.                                             *****
                   M4REPO   -      133 RECORDS INPUT
                   M4LIST   -       51 RECORDS OUTPUT
                   DEPTTOTS -       12 RECORDS OUTPUT
                   SUMRPTDD -       13 RECORDS OUTPUT
** MK4W209  TYPE 0  1057112 BYTES OF MAIN STORAGE UNUSED DURING REPORTING PHASE                     *****
```

Figure D-37 Report Summary File Specification (Page 4 of 4)

# Data Structure Handling

## Multi-Path Nested Loop in the Subfile Output Command Group

```
-----------------------------------------------------------------------------------------------------------------------
  RUN     STMT    FILE    O N S D U T R A B S D S L  RB  R S M R S            START           END             SEQUENCE   LIST
  NAME    TYPE    NAME    L E E I P R E U U R L C S  PL  J O O F S            SEARCH          SEARCH          73 TO 80    SEQ
                          D W Q R D N P D F T M N T  TK  T P P O R
-----------------------------------------------------------------------------------------------------------------------
(ASLRC   ) (RC) (CUSTOMER) (S)         (S)         (G)          (R)                                        (ASLGEN'D)     1
    CONTROL  LISTGEN
    FILE MASTER INPUT, NAME CUSTOMER
    FILE SUBF1 NAME CUSTSUB
    FILE REPORT
    ;
-----------------------------------------------------------------------------------------------------------------------
  RUN     STMT  LOGICAL  LBL  DTF/DD      KEY       KEY       KEY      QUALIFIER   USER I/O  ARRAY      PASSWORD/  SEQUENCE   LIST
  NAME    TYPE  FILENAME       NAME     FIELD-1   FIELD-2   FIELD-3   ICF CRD DBI   MODULE   QUAL       AUTH.-ID   73 TO 80    SEQ
-----------------------------------------------------------------------------------------------------------------------
(ASLRC   ) (RF) (CUSTOMER) (S) (M4OLD   )                                                                (ASLGEN'D)     2
(ASLRC   ) (RF) (CUSTSUB ) (S) (M4SUBF1 )                                                                (ASLGEN'D)     3


=================================================================================================================
=                                                                                                               =
=                                                  F I L E   S U M M A R Y                                      =
=                                                                                                               =
= JOBNAME: HANME028                          STEPNAME: F204231                                                  =
=                                                                                                               =
=  FILE NAME: M4OLD                          DSNAME: BUILDER.R140.CUSTOMER.MASTER.R08#1                         =
=  FILE NAME: M4SUBF1                        DSNAME: BUILDER.R140.M4SUBF1.R08#1                                 =
=  FILE NAME: M4LIB                          DSNAME: BUILDER.R140.M4LIB.R08#1                                   =
=                                                                                                               =
=================================================================================================================
NOV 15, 2001  12.34.58                                                                                  PAGE     3
                                                  ******************************
```

Figure D-38 Multi-Path Nested Loop in the Subfile Output Command Group (Page 1 of 12)

```
                                        *  PROC NAME - MAIN        *
                                        *  INPUT STREAM PROCEDURE  *
                                        ****************************
------------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                             MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ         BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID        ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM      73 TO 80    SEQ
------------------------------------------------------------------------------------------------------------------------------------
(ER)                                                                            (N)                          (ASLGEN'D)     4
    MAIN: PROC
    ;Each Master-File-Record Control Procedure
        CALL PROCLOOP
        CALL PROCLUP3
        CALL PRLOOP67
        CALL PRLOOP8
    END PROC
    ;
------------------------------------------------------------------------------------------------------------------------------------
STMT SEQ  LOG CON .....OPERAND-A.....  OPER  .............OPERAND-B.............   ......RESULT......  PARTIAL FIELD  SEQUENCE   LIST
TYPE NO.  LEV CTR QLF FIELD   SEG-LVL  ATN   QLF FIELD   SEG-LVL  (OR CONSTANT)   QLF FIELD  SEG-LVL  STRT NUMB OPR  73 TO 80    SEQ
------------------------------------------------------------------------------------------------------------------------------------
(PR) (100)                          (GO)     (SUB PROCLOOP                 )                                      (ASLGEN'D)     5
(PR) (101)                          (GO)     (SUB PROCLUP3                 )                                      (ASLGEN'D)     6
(PR) (102)                          (GO)     (SUB PRLOOP67                 )                                      (ASLGEN'D)     7
(PR) (103)                          (GO)     (SUB PRLOOP8                  )                                      (ASLGEN'D)     8
NOV 15, 2001  12.34.58                                                                                                  PAGE     4
                                        ****************************
                                        *  PROC NAME - SBNESTED    *
                                        *  INPUT STREAM PROCEDURE  *
                                        ****************************
------------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                             MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ         BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID        ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM      73 TO 80    SEQ
------------------------------------------------------------------------------------------------------------------------------------
(ER)                                                                            (N)                          (ASLGEN'D)     9
    SBNESTED: EXTRACT   FILE CUSTSUB,
              ITEMS CUSTNO, CUSTNAME, ORDERNO, ORDRDATE, ORPERSON,
              INSTNO, INSTOPS
    ; Create subfile using multi-path nested looping
    ;
------------------------------------------------------------------------------------------------------------------------------------
 ST SUM V  8  PG  PG   I SP  MAXIMUM TB COLUMN  DT  PG  STRT LIN SL RH .......SUBFILE........ C E A S R S N      C  SEQUENCE   LIST
 TY RPT S LPI WID HGT  M FRM LPP PGS TO HEADING POS POS PAGE NUM UB PN ENT    NAME   BLK/ FMT H F G T S U D      O  73 TO 80    SEQ
 MP     P            G            LT TYP POS         NUM      ML TD REC          BUF      C C S L T M S      N
------------------------------------------------------------------------------------------------------------------------------------
(E1)                                                        (NR)   (CUSTSUB )                                (ASLGEN'D)    10
** MK4FR75  TYPE 0  SUBFILE RECORD FORMAT IS DEFAULTED TO A V-TYPE.                                                      *****
------------------------------------------------------------------------------------------------------------------------------------
         SP  Q               E  N  S  D  C  S  T  C  C  M  M  A  G  D Q PCT/RAT     P                PARTIAL   SEQUENCE   LIST
STMT SEQ BF  L  FIELD   SEG-  N  P  E  E  T  U  O  U  N  A  I  V  M  E L  FIELD   RA- C     PICTURE     STRT NO  73 TO 80    SEQ
TYPE NO. COL F  NAME      LVL D  R  Q  S  L  B  T  M  T  X  N  G  R  C F  NAME    TIO T      EDIT       CHR CHR
------------------------------------------------------------------------------------------------------------------------------------
(R1)           (CUSTNO  ) 001-1                                                                                     (ASLGEN'D)    11
(R1)           (CUSTNAME) 001-1                                                                                     (ASLGEN'D)    12
(R1)           (ORDERNO ) 002-2                                                                                     (ASLGEN'D)    13
(R1)           (ORDRDATE) 002-2                                                                                     (ASLGEN'D)    14
(R1)           (ORPERSON) 002-2                                                                                     (ASLGEN'D)    15
(R1)           (INSTNO  ) 006-2                                                                                     (ASLGEN'D)    16
(R1)           (INSTOPS ) 006-2                                                                                     (ASLGEN'D)    17
** MK4TY05  TYPE 0  SUBFILE RECORD LENGTH COMPUTED FROM THE ABOVE SPECIFICATION IS    65.                                      *****
NOV 15, 2001  12.34.58                                                                                                  PAGE     5
                                        ****************************
                                        *  PROC NAME - SBPARALL    *
                                        *  INPUT STREAM PROCEDURE  *
                                        ****************************
------------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                             MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ         BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID        ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM      73 TO 80    SEQ
------------------------------------------------------------------------------------------------------------------------------------
(ER)                                                                            (N)                   (R)    (ASLGEN'D)    18
    SBPARALL: EXTRACT   FILE CUSTSUB, ABBREVIATED,
              ITEMS CUSTNO, CUSTNAME, ORDERNO, ORDRDATE, ORPERSON,
              INSTNO, INSTOPS
    ; Create subfile using multi-path parallel looping
    ;
------------------------------------------------------------------------------------------------------------------------------------
 ST SUM V  8  PG  PG   I SP  MAXIMUM TB COLUMN  DT  PG  STRT LIN SL RH .......SUBFILE........ C E A S R S N      C  SEQUENCE   LIST
 TY RPT S LPI WID HGT  M FRM LPP PGS TO HEADING POS POS PAGE NUM UB PN ENT    NAME   BLK/ FMT H F G T S U D      O  73 TO 80    SEQ
 MP     P            G            LT TYP POS         NUM      ML TD REC          BUF      C C S L T M S      N
------------------------------------------------------------------------------------------------------------------------------------
(E1)                                                        (NR)   (CUSTSUB )                                (ASLGEN'D)    19
** MK4FR75  TYPE 0  SUBFILE RECORD FORMAT IS DEFAULTED TO A V-TYPE.                                                      *****
------------------------------------------------------------------------------------------------------------------------------------
         SP  Q               E  N  S  D  C  S  T  C  C  M  M  A  G  D Q PCT/RAT     P                PARTIAL   SEQUENCE   LIST
STMT SEQ BF  L  FIELD   SEG-  N  P  E  E  T  U  O  U  N  A  I  V  M  E L  FIELD   RA- C     PICTURE     STRT NO  73 TO 80    SEQ
TYPE NO. COL F  NAME      LVL D  R  Q  S  L  B  T  M  T  X  N  G  R  C F  NAME    TIO T      EDIT       CHR CHR
```

Figure D-38 Multi-Path Nested Loop in the Subfile Output Command Group (Page 2 of 12)

```
--------------------------------------------------------------------------------------------------------------------------------
(R1)           (CUSTNO  ) 001-1                                                                                  (ASLGEN'D)    20
(R1)           (CUSTNAME) 001-1                                                                                  (ASLGEN'D)    21
(R1)           (ORDERNO ) 002-2                                                                                  (ASLGEN'D)    22
(R1)           (ORDRDATE) 002-2                                                                                  (ASLGEN'D)    23
(R1)           (ORPERSON) 002-2                                                                                  (ASLGEN'D)    24
(R1)           (INSTNO  ) 006-2                                                                                  (ASLGEN'D)    25
(R1)           (INSTOPS ) 006-2                                                                                  (ASLGEN'D)    26
** MK4TY05  TYPE 0  SUBFILE RECORD LENGTH COMPUTED FROM THE ABOVE SPECIFICATION IS    65.                                    *****
NOV 15, 2001  12.34.58                                                                                             PAGE      6
                                              ******************************
                                              *  PROC NAME - RPPARALL      *
                                              *  INPUT STREAM PROCEDURE    *
                                              ******************************
--------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                              MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID        ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM      73 TO 80    SEQ
--------------------------------------------------------------------------------------------------------------------------------
(ER)                                                                          (N)                    (R)      (ASLGEN'D)    27
    RPPARALL: REPORT  ABBREVIATED,
              ITEMS CUSTNO, CUSTNAME, ORDERNO, ORDRDATE, ORPERSON,
              INSTNO, INSTOPS
              TITLE 'SIMPLE REPORT USING MULTI-PATH PARALLEL LOOPING'
          END REPORT
    ;
--------------------------------------------------------------------------------------------------------------------------------
 ST SUM V  8  PG  PG   I SP  MAXIMUM TB COLUMN  DT  PG  STRT LIN SL RH .......SUBFILE..... C E A S R S N      C  SEQUENCE   LIST
 TY RPT S LPI WID HGT  M FRM LPP PGS TO HEADING POS POS PAGE NUM UB PN ENT    NAME   BLK/ FMT H F G T S U D      O  73 TO 80    SEQ
 MP     P             G            LT TYP POS         NUM      ML TD REC           BUF      C C S L T M S      N
--------------------------------------------------------------------------------------------------------------------------------
(E1)                                                                                                           (ASLGEN'D)    28
--------------------------------------------------------------------------------------------------------------------------------
         SP  Q               E N  S  D  C  S  T  C  C  M  M  A  G  D Q PCT/RAT     P              PARTIAL   SEQUENCE   LIST
STMT SEQ  BF  L  FIELD   SEG-  N P  E  E  T  U  O  U  N  A  I  V  M  E L  FIELD   RA- C     PICTURE     STRT NO   73 TO 80    SEQ
TYPE NO.  COL F  NAME      LVL  D R  Q  S  L  B  T  M  T  X  N  G  R  C F  NAME    TIO T      EDIT       CHR CHR
--------------------------------------------------------------------------------------------------------------------------------
(R1)           (CUSTNO  ) 001-1                                                                                  (ASLGEN'D)    29
(R1)           (CUSTNAME) 001-1                                                                                  (ASLGEN'D)    30
(R1)           (ORDERNO ) 002-2                                                                                  (ASLGEN'D)    31
(R1)           (ORDRDATE) 002-2                                                                                  (ASLGEN'D)    32
(R1)           (ORPERSON) 002-2                                                                                  (ASLGEN'D)    33
(R1)           (INSTNO  ) 006-2                                                                                  (ASLGEN'D)    34
(R1)           (INSTOPS ) 006-2                                                                                  (ASLGEN'D)    35
--------------------------------------------------------------------------------------------------------------------------------
STMT  SEQ                                                                                                          SEQUENCE   LIST
TYPE  NO.                                        TEXT                                                           73 TO 80    SEQ
--------------------------------------------------------------------------------------------------------------------------------
(T1) (101)                (SIMPLE REPORT USING MULTI-PATH PARALLEL LOOPING#          )                          (ASLGEN'D)    36
NOV 15, 2001  12.34.58                                                                                             PAGE      7
                                              ******************************
                                              *  PROC NAME - PROCLOOP      *
                                              *  INPUT STREAM PROCEDURE    *
                                              ******************************
--------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                              MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID        ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM      73 TO 80    SEQ
--------------------------------------------------------------------------------------------------------------------------------
(ER)                                                                          (S)                              (ASLGEN'D)    37
    PROCLOOP: PROC TYPE SUBROUTINE
        IF CUSTNO NE '00005'
    A020:  IF ORDRDATE EQ ORDRDATE
              CALL REPORT REPORT1  ;invokes report with looping
              CALL REPORT REPORT1A ;invokes report with limited range
           END
        END
    END
    ;
--------------------------------------------------------------------------------------------------------------------------------
STMT SEQ  LOG CON .....OPERAND-A.....  OPER  .............OPERAND-B.............   ......RESULT......  PARTIAL FIELD  SEQUENCE   LIST
TYPE NO.  LEV CTR QLF FIELD   SEG-LVL  ATN   QLF FIELD   SEG-LVL  (OR CONSTANT)   QLF FIELD  SEG-LVL  STRT NUMB OPR  73 TO 80    SEQ
--------------------------------------------------------------------------------------------------------------------------------
(PR) (100)          (CUSTNO  ) 001-1  (NE)    (C,00005                        )                                    (ASLGEN'D)    38
(PR) (101)                            (NS)      (END                            )                                    (ASLGEN'D)    39
(PR) (102)          (ORDRDATE) 002-2  (EQ)      (ORDRDATE) 002-2                                                     (ASLGEN'D)    40
(PR) (103)                            (NS)      (END                            )                                    (ASLGEN'D)    41
(PR) (104)                            (GO)      (SUB REPORT1                    )                                    (ASLGEN'D)    42
(PR) (105)                            (GO)      (SUB REPORT1A                   )                                    (ASLGEN'D)    43
NOV 15, 2001  12.34.58                                                                                             PAGE      8
                                              ******************************
                                              *  PROC NAME - REPORT1       *
                                              *  INPUT STREAM PROCEDURE    *
                                              ******************************
--------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                              MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID        ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM      73 TO 80    SEQ
```

Figure D-38 Multi-Path Nested Loop in the Subfile Output Command Group (Page 3 of 12)

```
------------------------------------------------------------------------------------------------------------------------
(ER)                                                                    (S)                          (ASLGEN'D)    44
    REPORT1: REPORT  TYPE SUBROUTINE,
             ITEMS CUSTNAME, CUSTNO, ORDERNO, ORDRDATE, ORPERSON
             TITLE 'LOOP DOMAIN LIMITED BY PROCEDURE PROCLOOP'
             END REPORT
    ;
------------------------------------------------------------------------------------------------------------------------
 ST SUM V  8  PG  PG   I SP  MAXIMUM TB COLUMN  DT  PG  STRT LIN SL RH .......SUBFILE........ C E A S R S N      C  SEQUENCE   LIST
 TY RPT S LPI WID HGT  M FRM LPP PGS TO HEADING POS POS PAGE NUM UB PN ENT    NAME   BLK/ FMT H F G T S U D      O  73 TO 80   SEQ
 MP     P             G           LT TYP POS        NUM      ML TD REC          BUF      C C S L T M S      N
------------------------------------------------------------------------------------------------------------------------
(E1)                                                                                                         (ASLGEN'D)    45
------------------------------------------------------------------------------------------------------------------------
        SP  Q              E N S  D  C  S  T  C  C  M  M  A  G  D Q PCT/RAT     P              PARTIAL   SEQUENCE   LIST
STMT SEQ BF  L  FIELD  SEG- N P E  E  T  U  O  U  N  A  I  V  M  E L  FIELD   RA- C     PICTURE    STRT NO   73 TO 80   SEQ
TYPE NO. COL F  NAME     LVL D R Q  S  L  B  T  M  T  X  N  G  R  C F  NAME    TIO T      EDIT      CHR CHR
------------------------------------------------------------------------------------------------------------------------
(R1)          (CUSTNAME) 001-1                                                                               (ASLGEN'D)    46
(R1)          (CUSTNO  ) 001-1                                                                               (ASLGEN'D)    47
(R1)          (ORDERNO ) 002-2                                                                               (ASLGEN'D)    48
(R1)          (ORDRDATE) 002-2                                                                               (ASLGEN'D)    49
(R1)          (ORPERSON) 002-2                                                                               (ASLGEN'D)    50
------------------------------------------------------------------------------------------------------------------------
STMT  SEQ                                                                                                    SEQUENCE   LIST
TYPE  NO.                                          TEXT                                                      73 TO 80   SEQ
------------------------------------------------------------------------------------------------------------------------
(T1) (101)              (LOOP DOMAIN LIMITED BY PROCEDURE PROCLOOP#              )                           (ASLGEN'D)    51
NOV 15, 2001  12.34.58                                                                                          PAGE     9
                                          ******************************
                                          *  PROC NAME - REPORT1A      *
                                          *  INPUT STREAM PROCEDURE    *
                                          ******************************
------------------------------------------------------------------------------------------------------------------------
STMT REPORT                              MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID        ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM       73 TO 80   SEQ
------------------------------------------------------------------------------------------------------------------------
(ER)                                                                    (S)                          (ASLGEN'D)    52
    REPORT1A: REPORT  TYPE SUBROUTINE,
              ITEMS CUSTNAME, CUSTNO
              TITLE 'LOOPS INITIATED IN PROCEDURE'
              TITLE 'RANGE LIMITED IN REPORT'
        END REPORT
    ;
------------------------------------------------------------------------------------------------------------------------
 ST SUM V  8  PG  PG   I SP  MAXIMUM TB COLUMN  DT  PG  STRT LIN SL RH .......SUBFILE........ C E A S R S N      C  SEQUENCE   LIST
 TY RPT S LPI WID HGT  M FRM LPP PGS TO HEADING POS POS PAGE NUM UB PN ENT    NAME   BLK/ FMT H F G T S U D      O  73 TO 80   SEQ
 MP     P             G           LT TYP POS        NUM      ML TD REC          BUF      C C S L T M S      N
------------------------------------------------------------------------------------------------------------------------
(E1)                                                                                                         (ASLGEN'D)    53
------------------------------------------------------------------------------------------------------------------------
        SP  Q              E N S  D  C  S  T  C  C  M  M  A  G  D Q PCT/RAT     P              PARTIAL   SEQUENCE   LIST
STMT SEQ BF  L  FIELD  SEG- N P E  E  T  U  O  U  N  A  I  V  M  E L  FIELD   RA- C     PICTURE    STRT NO   73 TO 80   SEQ
TYPE NO. COL F  NAME     LVL D R Q  S  L  B  T  M  T  X  N  G  R  C F  NAME    TIO T      EDIT      CHR CHR
------------------------------------------------------------------------------------------------------------------------
(R1)          (CUSTNAME) 001-1                                                                               (ASLGEN'D)    54
(R1)          (CUSTNO  ) 001-1                                                                               (ASLGEN'D)    55
------------------------------------------------------------------------------------------------------------------------
STMT  SEQ                                                                                                    SEQUENCE   LIST
TYPE  NO.                                          TEXT                                                      73 TO 80   SEQ
------------------------------------------------------------------------------------------------------------------------
(T1) (101)              (LOOPS INITIATED IN PROCEDURE#                           )                          (ASLGEN'D)    56
(T1) (102)              (RANGE LIMITED IN REPORT#                                )                          (ASLGEN'D)    57
NOV 15, 2001  12.34.58                                                                                          PAGE     10
                                          ******************************
                                          *  PROC NAME - PROCLUP3      *
                                          *  INPUT STREAM PROCEDURE    *
                                          ******************************
------------------------------------------------------------------------------------------------------------------------
STMT REPORT                              MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID        ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM       73 TO 80   SEQ
------------------------------------------------------------------------------------------------------------------------
(ER)                                                                    (S)                          (ASLGEN'D)    58
    PROCLUP3: PROC TYPE SUBROUTINE
       IF CUSTNO NE '00005' AND CUSTNO NE '00015'
            CALL REPORT REPORT2 ;  Only the root segment is referenced
            CALL REPORT REPORT4 ;  in the procedure.  All loops are
            CALL REPORT REPORT5 ;  initiated in the reports.
       END
    END
    ;
------------------------------------------------------------------------------------------------------------------------
STMT SEQ  LOG CON .....OPERAND-A.....  OPER  .............OPERAND-B.............  ......RESULT......  PARTIAL FIELD  SEQUENCE   LIST
TYPE NO.  LEV CTR QLF FIELD   SEG-LVL  ATN   QLF FIELD   SEG-LVL  (OR CONSTANT)  QLF FIELD  SEG-LVL  STRT NUMB OPR  73 TO 80   SEQ
------------------------------------------------------------------------------------------------------------------------
```

Figure D-38 Multi-Path Nested Loop in the Subfile Output Command Group (Page 4 of 12)

```
(PR) (100)          (CUSTNO  ) 001-1  (NE)   (C,00005                    )                    (ASLGEN'D)    59
(PR) (101)                            (NS)     (END                       )                    (ASLGEN'D)    60
(PR) (102)          (CUSTNO  ) 001-1  (NE)   (C,00015                    )                    (ASLGEN'D)    61
(PR) (103)                            (NS)     (END                       )                    (ASLGEN'D)    62
(PR) (104)                            (GO)     (SUB REPORT2               )                    (ASLGEN'D)    63
(PR) (105)                            (GO)     (SUB REPORT4               )                    (ASLGEN'D)    64
(PR) (106)                            (GO)     (SUB REPORT5               )                    (ASLGEN'D)    65
NOV 15, 2001  12.34.58                                                                              PAGE    11
                                                 ****************************
                                                 *  PROC NAME - REPORT2     *
                                                 *  INPUT STREAM PROCEDURE  *
                                                 ****************************

-------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                              MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID        ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM       73 TO 80   SEQ
-------------------------------------------------------------------------------------------------------------------------------
(ER)                                                                          (S)                          (ASLGEN'D)    66
    REPORT2: REPORT  TYPE SUBROUTINE,
             ITEMS CUSTNAME, CUSTNO, ORDERNO, ORDRDATE, ORPERSON
             TITLE 'LOOP STARTS IN REPORT'
             END REPORT
    ;
-------------------------------------------------------------------------------------------------------------------------------
 ST SUM V  8  PG  PG   I SP  MAXIMUM TB COLUMN  DT  PG  STRT LIN SL RH .......SUBFILE..... C E A S R S N      C  SEQUENCE   LIST
 TY RPT S LPI WID HGT  M FRM LPP PGS TO HEADING POS POS PAGE NUM UB PN ENT    NAME   BLK/ FMT H F G T S U D      O  73 TO 80   SEQ
 MP     P            G            LT TYP POS        NUM      ML TD REC           BUF      C C S L T M S      N
-------------------------------------------------------------------------------------------------------------------------------
(E1)                                                                                                         (ASLGEN'D)    67
-------------------------------------------------------------------------------------------------------------------------------
        SP  Q              E N S D C S T C C M M A G D Q PCT/RAT     P              PARTIAL   SEQUENCE   LIST
STMT SEQ BF  L FIELD  SEG-  N P E E T U O U N A I V M E L  FIELD   RA- C    PICTURE    STRT NO  73 TO 80   SEQ
TYPE NO. COL F NAME     LVL D R Q S L B T M T X N G R C F  NAME    TIO T     EDIT      CHR CHR
-------------------------------------------------------------------------------------------------------------------------------
(R1)          (CUSTNAME) 001-1                                                                               (ASLGEN'D)    68
(R1)          (CUSTNO  ) 001-1                                                                               (ASLGEN'D)    69
(R1)          (ORDERNO ) 002-2                                                                               (ASLGEN'D)    70
(R1)          (ORDRDATE) 002-2                                                                               (ASLGEN'D)    71
(R1)          (ORPERSON) 002-2                                                                               (ASLGEN'D)    72
-------------------------------------------------------------------------------------------------------------------------------
STMT  SEQ                                                                                                    SEQUENCE   LIST
TYPE  NO.                                        TEXT                                                        73 TO 80   SEQ
-------------------------------------------------------------------------------------------------------------------------------
(T1) (101)              (LOOP STARTS IN REPORT#                                  )                           (ASLGEN'D)    73
NOV 15, 2001  12.34.58                                                                                        PAGE    12
                                                 ****************************
                                                 *  PROC NAME - REPORT4     *
                                                 *  INPUT STREAM PROCEDURE  *
                                                 ****************************

-------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                              MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID        ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM       73 TO 80   SEQ
-------------------------------------------------------------------------------------------------------------------------------
(ER)                                                                          (S)                          (ASLGEN'D)    74
    REPORT4: REPORT  TYPE SUBROUTINE,
             ITEMS CUSTNAME, CUSTNO, ORDERNO, ORDRDATE, ORPERSON,
                   SHIPNO, SHIPDATE
             TITLE 'SINGLE-PATH NESTED LOOP STARTED IN REPORT'
             TITLE 'HIERARCHIACAL ORDER'
             END REPORT
    ;
-------------------------------------------------------------------------------------------------------------------------------
 ST SUM V  8  PG  PG   I SP  MAXIMUM TB COLUMN  DT  PG  STRT LIN SL RH .......SUBFILE..... C E A S R S N      C  SEQUENCE   LIST
 TY RPT S LPI WID HGT  M FRM LPP PGS TO HEADING POS POS PAGE NUM UB PN ENT    NAME   BLK/ FMT H F G T S U D      O  73 TO 80   SEQ
 MP     P            G            LT TYP POS        NUM      ML TD REC           BUF      C C S L T M S      N
-------------------------------------------------------------------------------------------------------------------------------
(E1)                                                                                                         (ASLGEN'D)    75
-------------------------------------------------------------------------------------------------------------------------------
        SP  Q              E N S D C S T C C M M A G D Q PCT/RAT     P              PARTIAL   SEQUENCE   LIST
STMT SEQ BF  L FIELD  SEG-  N P E E T U O U N A I V M E L  FIELD   RA- C    PICTURE    STRT NO  73 TO 80   SEQ
TYPE NO. COL F NAME     LVL D R Q S L B T M T X N G R C F  NAME    TIO T     EDIT      CHR CHR
-------------------------------------------------------------------------------------------------------------------------------
(R1)          (CUSTNAME) 001-1                                                                               (ASLGEN'D)    76
(R1)          (CUSTNO  ) 001-1                                                                               (ASLGEN'D)    77
(R1)          (ORDERNO ) 002-2                                                                               (ASLGEN'D)    78
(R1)          (ORDRDATE) 002-2                                                                               (ASLGEN'D)    79
(R1)          (ORPERSON) 002-2                                                                               (ASLGEN'D)    80
(R1)          (SHIPNO  ) 003-3                                                                               (ASLGEN'D)    81
(R1)          (SHIPDATE) 003-3                                                                               (ASLGEN'D)    82
-------------------------------------------------------------------------------------------------------------------------------
STMT  SEQ                                                                                                    SEQUENCE   LIST
TYPE  NO.                                        TEXT                                                        73 TO 80   SEQ
-------------------------------------------------------------------------------------------------------------------------------
(T1) (101)              (SINGLE-PATH NESTED LOOP STARTED IN REPORT#               )                           (ASLGEN'D)    83
(T1) (102)              (HIERARCHIACAL ORDER#                                    )                           (ASLGEN'D)    84
NOV 15, 2001  12.34.58                                                                                        PAGE    13
```

Figure D-38 Multi-Path Nested Loop in the Subfile Output Command Group (Page 5 of 12)

```
                                        ****************************
                                        *  PROC NAME - REPORT5     *
                                        *  INPUT STREAM PROCEDURE  *
                                        ****************************
-------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                              MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID        ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM      73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(ER)                                                                            (S)                          (ASLGEN'D)    85
    REPORT5: REPORT  TYPE SUBROUTINE,
             ITEMS SHIPNO, SHIPDATE,
                   ORDERNO, ORDRDATE, ORPERSON, CUSTNAME, CUSTNO
             TITLE 'SINGLE-PATH NESTED LOOP STARTED IN REPORT'
             TITLE 'INVERSE HIERARCHIACAL ORDER'
             END REPORT
    ;
-------------------------------------------------------------------------------------------------------------------------------
 ST SUM V  8  PG  PG   I SP  MAXIMUM TB COLUMN  DT  PG  STRT LIN SL RH .......SUBFILE........ C E A S R S N      C  SEQUENCE   LIST
 TY RPT S LPI WID HGT  M FRM LPP PGS TO HEADING POS POS PAGE NUM UB PN ENT    NAME    BLK/ FMT H F G T S U D      O  73 TO 80    SEQ
 MP     P             G            LT TYP POS         NUM      ML TD REC           BUF      C C S L T M S      N
-------------------------------------------------------------------------------------------------------------------------------
(E1)                                                                                                          (ASLGEN'D)    86
-------------------------------------------------------------------------------------------------------------------------------
         SP  Q               E  N  S  D  C  S  T  C  C  M  M  A  G  D Q PCT/RAT     P                PARTIAL   SEQUENCE   LIST
STMT SEQ BF  L  FIELD   SEG-  N  P  E  E  T  U  O  U  N  A  I  V  M  E L  FIELD   RA- C     PICTURE     STRT NO  73 TO 80    SEQ
TYPE NO. COL F  NAME     LVL  D  R  Q  S  L  B  T  M  T  X  N  G  R  C F  NAME    TIO T      EDIT      CHR CHR
-------------------------------------------------------------------------------------------------------------------------------
(R1)          (SHIPNO  ) 003-3                                                                            (ASLGEN'D)    87
(R1)          (SHIPDATE) 003-3                                                                            (ASLGEN'D)    88
(R1)          (ORDERNO ) 002-2                                                                            (ASLGEN'D)    89
(R1)          (ORDRDATE) 002-2                                                                            (ASLGEN'D)    90
(R1)          (ORPERSON) 002-2                                                                            (ASLGEN'D)    91
(R1)          (CUSTNAME) 001-1                                                                            (ASLGEN'D)    92
(R1)          (CUSTNO  ) 001-1                                                                            (ASLGEN'D)    93
-------------------------------------------------------------------------------------------------------------------------------
STMT  SEQ                                                                                                    SEQUENCE   LIST
TYPE  NO.                                        TEXT                                                        73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(T1) (101)                (SINGLE-PATH NESTED LOOP STARTED IN REPORT#              )                          (ASLGEN'D)    94
(T1) (102)                (INVERSE HIERARCHIACAL ORDER#                           )                          (ASLGEN'D)    95
NOV 15, 2001  12.34.58                                                                                          PAGE    14
                                        ****************************
                                        *  PROC NAME - PRLOOP67    *
                                        *  INPUT STREAM PROCEDURE  *
                                        ****************************
-------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                              MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID        ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM      73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(ER)                                                                            (S)                          (ASLGEN'D)    96
    PRLOOP67: PROC TYPE SUBROUTINE
       IF CUSTNO NE '00005' AND CUSTNO NE '00015'
            CALL REPORT REPORT6  ;  report uses multi path nested loops
            CALL REPORT REPORT7  ;  report uses multi path parallel loops
       END
    END
    ;
-------------------------------------------------------------------------------------------------------------------------------
STMT SEQ  LOG CON .....OPERAND-A.....  OPER  .............OPERAND-B.............   .......RESULT.......  PARTIAL FIELD  SEQUENCE   LIST
TYPE NO.  LEV CTR QLF FIELD   SEG-LVL  AIN   QLF FIELD   SEG-LVL  (OR CONSTANT)   QLF FIELD  SEG-LVL  STRT NUMB OPR  73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(PR) (100)          (CUSTNO  ) 001-1  (NE)   (C,00005                        )                                    (ASLGEN'D)    97
(PR) (101)                            (NS)     (END                          )                                    (ASLGEN'D)    98
(PR) (102)          (CUSTNO  ) 001-1  (NE)   (C,00015                        )                                    (ASLGEN'D)    99
(PR) (103)                            (NS)     (END                          )                                    (ASLGEN'D)   100
(PR) (104)                            (GO)     (SUB REPORT6                  )                                    (ASLGEN'D)   101
(PR) (105)                            (GO)     (SUB REPORT7                  )                                    (ASLGEN'D)   102
NOV 15, 2001  12.34.58                                                                                          PAGE    15
                                        ****************************
                                        *  PROC NAME - REPORT6     *
                                        *  INPUT STREAM PROCEDURE  *
                                        ****************************
-------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                              MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID        ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM      73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(ER)                                                                            (S)                          (ASLGEN'D)   103
    REPORT6: REPORT  TYPE SUBROUTINE,
             ITEMS CUSTNAME, CUSTNO, ORDERNO, ORPERSON,
                   INSTNO, INSTOPS
             ORDER BY CUSTNAME, CUSTNO
             GROUP BY CUSTNAME, CUSTNO
             TITLE 'MULTI-PATH NESTED LOOPING'
             END REPORT
    ;
```

Figure D-38 Multi-Path Nested Loop in the Subfile Output Command Group (Page 6 of 12)

```
------------------------------------------------------------------------------------------------------------------------------------
 ST SUM V  8  PG  PG   I SP  MAXIMUM TB COLUMN  DT  PG  STRT LIN SL RH .......SUBFILE..... C E A S R S N      C  SEQUENCE   LIST
 TY RPT S LPI WID HGT  M FRM LPP PGS TO HEADING POS POS PAGE NUM UB PN ENT    NAME   BLK/ FMT H F G T S U D      O  73 TO 80    SEQ
 MP     P             G           LT TYP POS        NUM      ML TD REC           BUF      C C S L T M S      N
------------------------------------------------------------------------------------------------------------------------------------
(E1)                                                                                                          (ASLGEN'D)   104
------------------------------------------------------------------------------------------------------------------------------------
        SP  Q              E N S  D C  S  T  C  C  M  M  A  G  D Q PCT/RAT     P              PARTIAL   SEQUENCE   LIST
STMT SEQ BF  L  FIELD   SEG-  N P  E  E  T  U  O  U  N  A  I  V  M  E L  FIELD  RA- C      PICTURE     STRT NO  73 TO 80    SEQ
TYPE NO. COL F  NAME     LVL  D R  Q  S  L  B  T  M  T  X  N  G  R  C F  NAME   TIO T       EDIT       CHR CHR
------------------------------------------------------------------------------------------------------------------------------------
(R1)          (CUSTNAME) 001-1      (1)   (1)                                                           (ASLGEN'D)   105
(R1)          (CUSTNO  ) 001-1      (2)   (1)                                                           (ASLGEN'D)   106
(R1)          (ORDERNO ) 002-2                                                                          (ASLGEN'D)   107
(R1)          (ORPERSON) 002-2                                                                          (ASLGEN'D)   108
(R1)          (INSTNO  ) 006-2                                                                          (ASLGEN'D)   109
(R1)          (INSTOPS ) 006-2                                                                          (ASLGEN'D)   110
------------------------------------------------------------------------------------------------------------------------------------
STMT  SEQ                                                                                               SEQUENCE   LIST
TYPE  NO.                                          TEXT                                                 73 TO 80    SEQ
------------------------------------------------------------------------------------------------------------------------------------
(T1) (101)                (MULTI-PATH NESTED LOOPING#                          )                        (ASLGEN'D)   111
NOV 15, 2001  12.34.58                                                                                     PAGE     16

                                                ****************************
                                                *  PROC NAME - REPORT7     *
                                                *  INPUT STREAM PROCEDURE  *
                                                ****************************
------------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                              MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID        ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM       73 TO 80    SEQ
------------------------------------------------------------------------------------------------------------------------------------
(ER)                                                                          (S)                  (R)      (ASLGEN'D)   112
    REPORT7: REPORT  TYPE SUBROUTINE, ABBREVIATED,
             ITEMS CUSTNAME, CUSTNO, ORDERNO, ORPERSON,
                   INSTNO, INSTOPS
             ORDER BY CUSTNAME, CUSTNO
             GROUP BY CUSTNAME, CUSTNO
             TITLE 'MULTI-PATH PARALLEL LOOPING'
             END REPORT
    ;
------------------------------------------------------------------------------------------------------------------------------------
 ST SUM V  8  PG  PG   I SP  MAXIMUM TB COLUMN  DT  PG  STRT LIN SL RH .......SUBFILE..... C E A S R S N      C  SEQUENCE   LIST
 TY RPT S LPI WID HGT  M FRM LPP PGS TO HEADING POS POS PAGE NUM UB PN ENT    NAME   BLK/ FMT H F G T S U D      O  73 TO 80    SEQ
 MP     P             G           LT TYP POS        NUM      ML TD REC           BUF      C C S L T M S      N
------------------------------------------------------------------------------------------------------------------------------------
(E1)                                                                                                          (ASLGEN'D)   113
------------------------------------------------------------------------------------------------------------------------------------
        SP  Q              E N S  D C  S  T  C  C  M  M  A  G  D Q PCT/RAT     P              PARTIAL   SEQUENCE   LIST
STMT SEQ BF  L  FIELD   SEG-  N P  E  E  T  U  O  U  N  A  I  V  M  E L  FIELD  RA- C      PICTURE     STRT NO  73 TO 80    SEQ
TYPE NO. COL F  NAME     LVL  D R  Q  S  L  B  T  M  T  X  N  G  R  C F  NAME   TIO T       EDIT       CHR CHR
------------------------------------------------------------------------------------------------------------------------------------
(R1)          (CUSTNAME) 001-1      (1)   (1)                                                           (ASLGEN'D)   114
(R1)          (CUSTNO  ) 001-1      (2)   (1)                                                           (ASLGEN'D)   115
(R1)          (ORDERNO ) 002-2                                                                          (ASLGEN'D)   116
(R1)          (ORPERSON) 002-2                                                                          (ASLGEN'D)   117
(R1)          (INSTNO  ) 006-2                                                                          (ASLGEN'D)   118
(R1)          (INSTOPS ) 006-2                                                                          (ASLGEN'D)   119
------------------------------------------------------------------------------------------------------------------------------------
STMT  SEQ                                                                                               SEQUENCE   LIST
TYPE  NO.                                          TEXT                                                 73 TO 80    SEQ
------------------------------------------------------------------------------------------------------------------------------------
(T1) (101)                (MULTI-PATH PARALLEL LOOPING#                        )                        (ASLGEN'D)   120
NOV 15, 2001  12.34.58                                                                                     PAGE     17

                                                ****************************
                                                *  PROC NAME - PRLOOP8     *
                                                *  INPUT STREAM PROCEDURE  *
                                                ****************************
------------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                              MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID        ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM       73 TO 80    SEQ
------------------------------------------------------------------------------------------------------------------------------------
(ER)                                                                          (S)                          (ASLGEN'D)   121
    PRLOOP8: PROC TYPE SUBROUTINE
        IF CUSTNO NE '00005' AND CUSTNO NE '00015'
          IF ORDERNO EQ ORDERNO       ; nested loop initiated here
              CALL REPORT REPORT8     ; will override 'ABBREVIATED'
          END                         ; parameter in REPORT8
        END
    END
------------------------------------------------------------------------------------------------------------------------------------
STMT SEQ  LOG CON .....OPERAND-A.....  OPER  .............OPERAND-B.............  .....RESULT.....  PARTIAL FIELD  SEQUENCE   LIST
TYPE NO.  LEV CTR QLF FIELD   SEG-LVL  ATN   QLF FIELD   SEG-LVL  (OR CONSTANT)  QLF FIELD  SEG-LVL  STRT NUMB OPR  73 TO 80    SEQ
------------------------------------------------------------------------------------------------------------------------------------
(PR) (100)          (CUSTNO  ) 001-1  (NE)   (C,00005                      )                                      (ASLGEN'D)   122
(PR) (101)                            (NS)     (END                        )                                      (ASLGEN'D)   123
```

Figure D-38 Multi-Path Nested Loop in the Subfile Output Command Group (Page 7 of 12)

```
(PR) (102)          (CUSTNO  ) 001-1  (NE)   (C,00015                    )                    (ASLGEN'D)   124
(PR) (103)                            (NS)     (END                        )                    (ASLGEN'D)   125
(PR) (104)          (ORDERNO ) 002-2  (EQ)     (ORDERNO ) 002-2                                 (ASLGEN'D)   126
(PR) (105)                            (NS)     (END                        )                    (ASLGEN'D)   127
(PR) (106)                            (GO)     (SUB REPORT8                )                    (ASLGEN'D)   128
NOV 15, 2001  12.34.58                                                                               PAGE    18

                                                ******************************
                                                *  PROC NAME - REPORT8       *
                                                *  INPUT STREAM PROCEDURE    *
                                                ******************************

-------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                              MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID        ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM      73 TO 80   SEQ
-------------------------------------------------------------------------------------------------------------------------------
(ER)                                                                          (S)                  (R)      (ASLGEN'D)   129
    REPORT8: REPORT  TYPE SUBROUTINE, ABBREVIATED,
             ITEMS CUSTNAME, CUSTNO, ORDERNO, ORPERSON,
                   INSTNO, INSTOPS
             ORDER BY CUSTNAME, CUSTNO
             GROUP BY CUSTNAME, CUSTNO
             TITLE 'MULTI-PATH PARALLEL LOOPING'
             END REPORT
    ;same report command group as report7 - looping initiated in
    ;invoking procedure PRLOOP8 is not overridden by ABBREVIATED parm
    ;
-------------------------------------------------------------------------------------------------------------------------------
 ST SUM V  8  PG  PG   I SP  MAXIMUM TB COLUMN  DT  PG  STRT LIN SL RH .......SUBFILE........ C E A S R S N      C  SEQUENCE   LIST
 TY RPT S LPI WID HGT  M FRM LPP PGS TO HEADING POS POS PAGE NUM UB PN ENT    NAME   BLK/ FMT H F G T S U D      O  73 TO 80   SEQ
 MP     P             G           LT TYP POS         NUM      ML TD REC           BUF      C C S L T M S      N
-------------------------------------------------------------------------------------------------------------------------------
(E1)                                                                                                         (ASLGEN'D)   130
-------------------------------------------------------------------------------------------------------------------------------
        SP  Q              E N S  D  C  S  T  C  C  M  M  A  G  D Q PCT/RAT     P              PARTIAL   SEQUENCE   LIST
STMT SEQ BF  L  FIELD   SEG- N P E  E  T  U  O  U  N  A  I  V  M  E L  FIELD   RA- C     PICTURE    STRT NO   73 TO 80   SEQ
TYPE NO. COL F  NAME     LVL D R Q  S  L  B  T  M  T  X  N  G  R  C F  NAME    TIO T      EDIT      CHR CHR
-------------------------------------------------------------------------------------------------------------------------------
(R1)          (CUSTNAME) 001-1      (1)   (1)                                                        (ASLGEN'D)   131
(R1)          (CUSTNO  ) 001-1      (2)   (1)                                                        (ASLGEN'D)   132
(R1)          (ORDERNO ) 002-2                                                                       (ASLGEN'D)   133
(R1)          (ORPERSON) 002-2                                                                       (ASLGEN'D)   134
(R1)          (INSTNO  ) 006-2                                                                       (ASLGEN'D)   135
(R1)          (INSTOPS ) 006-2                                                                       (ASLGEN'D)   136
-------------------------------------------------------------------------------------------------------------------------------
STMT  SEQ                                                                                            SEQUENCE   LIST
TYPE  NO.                                          TEXT                                              73 TO 80   SEQ
-------------------------------------------------------------------------------------------------------------------------------
(T1) (101)                 (MULTI-PATH PARALLEL LOOPING#                          )                  (ASLGEN'D)   137
NOV 15, 2001  12.34.58                                                                               PAGE    19

                                                ******************************
                                                *  PROC NAME - SELECT        *
                                                *  INPUT STREAM PROCEDURE    *
                                                ******************************

-------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                              MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID        ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM      73 TO 80   SEQ
-------------------------------------------------------------------------------------------------------------------------------
(ER)                                                 (Y)                      (N)                     (ASLGEN'D)   138
    SELECT: REPORT  SELECTION CONTROL YES,
             ITEMS CUSTNAME, CUSTNO, ORDERNO, ORPERSON,
                   INSTNO, INSTOPS
             ORDER BY CUSTNAME, CUSTNO
             GROUP BY CUSTNAME, CUSTNO
             TITLE 'LOOPING LIMITED BY SELECTION CONTROL'
             END REPORT
    ;
-------------------------------------------------------------------------------------------------------------------------------
 ST SUM V  8  PG  PG   I SP  MAXIMUM TB COLUMN  DT  PG  STRT LIN SL RH .......SUBFILE........ C E A S R S N      C  SEQUENCE   LIST
 TY RPT S LPI WID HGT  M FRM LPP PGS TO HEADING POS POS PAGE NUM UB PN ENT    NAME   BLK/ FMT H F G T S U D      O  73 TO 80   SEQ
 MP     P             G           LT TYP POS         NUM      ML TD REC           BUF      C C S L T M S      N
-------------------------------------------------------------------------------------------------------------------------------
(E1)                                                                                                         (ASLGEN'D)   139
-------------------------------------------------------------------------------------------------------------------------------
        SP  Q              E N S  D  C  S  T  C  C  M  M  A  G  D Q PCT/RAT     P              PARTIAL   SEQUENCE   LIST
STMT SEQ BF  L  FIELD   SEG- N P E  E  T  U  O  U  N  A  I  V  M  E L  FIELD   RA- C     PICTURE    STRT NO   73 TO 80   SEQ
TYPE NO. COL F  NAME     LVL D R Q  S  L  B  T  M  T  X  N  G  R  C F  NAME    TIO T      EDIT      CHR CHR
-------------------------------------------------------------------------------------------------------------------------------
(R1)          (CUSTNAME) 001-1      (1)   (1)                                                        (ASLGEN'D)   140
(R1)          (CUSTNO  ) 001-1      (2)   (1)                                                        (ASLGEN'D)   141
(R1)          (ORDERNO ) 002-2                                                                       (ASLGEN'D)   142
(R1)          (ORPERSON) 002-2                                                                       (ASLGEN'D)   143
(R1)          (INSTNO  ) 006-2                                                                       (ASLGEN'D)   144
(R1)          (INSTOPS ) 006-2                                                                       (ASLGEN'D)   145
-------------------------------------------------------------------------------------------------------------------------------
STMT  SEQ                                                                                            SEQUENCE   LIST
TYPE  NO.                                          TEXT                                              73 TO 80   SEQ
```

Figure D-38 Multi-Path Nested Loop in the Subfile Output Command Group (Page 8 of 12)

```
------------------------------------------------------------------------------------------------------------------------------
(T1) (101)                  (LOOPING LIMITED BY SELECTION CONTROL#                    )                          (ASLGEN'D)   146
NOV 15, 2001  12.34.58                                                                                             PAGE     20
                                          ******************************
                                          *  PROC NAME - FINDWH        *
                                          *  INPUT STREAM PROCEDURE    *
                                          ******************************
------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                              MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID        ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM       73 TO 80    SEQ
------------------------------------------------------------------------------------------------------------------------------
(ER)                                                                                 (N)                           (ASLGEN'D)   147
    FINDWH: PROC
          IF FIND(ORDER WHERE ORDERNO = '01104')
              CALL REPORT REPORT9
          END
    END
------------------------------------------------------------------------------------------------------------------------------
STMT SEQ  LOG CON .....OPERAND-A.....  OPER  .............OPERAND-B.............   ......RESULT......  PARTIAL FIELD  SEQUENCE   LIST
TYPE NO.  LEV CTR QLF FIELD   SEG-LVL  ATN   QLF FIELD   SEG-LVL  (OR CONSTANT)   QLF FIELD  SEG-LVL  STRT NUMB OPR  73 TO 80    SEQ
------------------------------------------------------------------------------------------------------------------------------
(PR) (100)          (ORDER   ) 002-2  (FS)    (C,01104                        )                                    (ASLGEN'D)   148
(PR) (101)                            (NS)      (END                           )                                    (ASLGEN'D)   149
(PR) (102)                            (GO)      (SUB REPORT9                   )                                    (ASLGEN'D)   150
NOV 15, 2001  12.34.58                                                                                             PAGE     21
                                          ******************************
                                          *  PROC NAME - REPORT9       *
                                          *  INPUT STREAM PROCEDURE    *
                                          ******************************
------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                              MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID        ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM       73 TO 80    SEQ
------------------------------------------------------------------------------------------------------------------------------
(ER)                                                                                 (S)                           (ASLGEN'D)   151
    REPORT9: REPORT TYPE SUBROUTINE,
             ITEMS CUSTNAME, CUSTNO, ORDERNO, ORPERSON, ORDRDATE
             TITLE 'FIND ORDER 01104'
             END REPORT
    ;
------------------------------------------------------------------------------------------------------------------------------
 ST SUM V  8  PG  PG   I SP  MAXIMUM TB COLUMN  DT  PG  STRT LIN SL RH .......SUBFILE........ C E A S R S N      C  SEQUENCE   LIST
 TY RPT S LPI WID HGT  M FRM LPP PGS TO HEADING POS POS PAGE NUM UB PN ENT    NAME   BLK/ FMT H F G T S U D      O  73 TO 80    SEQ
 MP     P             G            LT TYP POS        NUM      ML TD REC           BUF      C C S L T M S      N
------------------------------------------------------------------------------------------------------------------------------
(E1)                                                                                                           (ASLGEN'D)   152
------------------------------------------------------------------------------------------------------------------------------
         SP  Q                E N  S  D  C  S  T  C  C  M  M  A  G  D Q PCT/RAT     P                PARTIAL   SEQUENCE   LIST
STMT SEQ BF  L  FIELD   SEG-   N P  E  E  T  U  O  U  N  A  I  V  M  E L  FIELD   RA- C     PICTURE    STRT NO   73 TO 80    SEQ
TYPE NO. COL F  NAME      LVL  D R  Q  S  L  B  T  M  T  X  N  G  R  C F  NAME    TIO T      EDIT      CHR CHR
------------------------------------------------------------------------------------------------------------------------------
(R1)           (CUSTNAME) 001-1                                                                                (ASLGEN'D)   153
(R1)           (CUSTNO  ) 001-1                                                                                (ASLGEN'D)   154
(R1)           (ORDERNO ) 002-2                                                                                (ASLGEN'D)   155
(R1)           (ORPERSON) 002-2                                                                                (ASLGEN'D)   156
(R1)           (ORDRDATE) 002-2                                                                                (ASLGEN'D)   157
------------------------------------------------------------------------------------------------------------------------------
STMT  SEQ                                                                                                       SEQUENCE   LIST
TYPE  NO.                                          TEXT                                                         73 TO 80    SEQ
------------------------------------------------------------------------------------------------------------------------------
(T1) (101)                  (FIND ORDER 01104#                                       )                          (ASLGEN'D)   158
NOV 15, 2001  12.34.58                                                                                             PAGE     22
                                          ******************************
                                          *  PROC NAME - FINDUNQL      *
                                          *  INPUT STREAM PROCEDURE    *
                                          ******************************
------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                              MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID        ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM       73 TO 80    SEQ
------------------------------------------------------------------------------------------------------------------------------
(ER)                                                                                 (N)                           (ASLGEN'D)   159
    FINDUNQL: PROC
          IF FIND(SEGMENT ORDER)
              CALL REPORT REP10
          END
    END
------------------------------------------------------------------------------------------------------------------------------
STMT SEQ  LOG CON .....OPERAND-A.....  OPER  .............OPERAND-B.............   ......RESULT......  PARTIAL FIELD  SEQUENCE   LIST
TYPE NO.  LEV CTR QLF FIELD   SEG-LVL  ATN   QLF FIELD   SEG-LVL  (OR CONSTANT)   QLF FIELD  SEG-LVL  STRT NUMB OPR  73 TO 80    SEQ
------------------------------------------------------------------------------------------------------------------------------
(PR) (100)          (ORDER   ) 002-2  (FS)                                                                         (ASLGEN'D)   160
(PR) (101)                            (NS)      (END                           )                                    (ASLGEN'D)   161
(PR) (102)                            (GO)      (SUB REP10                     )                                    (ASLGEN'D)   162
NOV 15, 2001  12.34.58                                                                                             PAGE     23
                                          ******************************
                                          *  PROC NAME - REP10         *
```

Figure D-38 Multi-Path Nested Loop in the Subfile Output Command Group (Page 9 of 12)

```
                                        *  INPUT STREAM PROCEDURE   *
                                        ****************************
-------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                              MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB      SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID        ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM      73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(ER)                                                                          (S)                          (ASLGEN'D)   163
    REP10: REPORT TYPE SUBROUTINE,
           ITEMS CUSTNAME, CUSTNO, ORDERNO, ORPERSON, ORDRDATE
           TITLE 'UNQUALIFIED FIND'
           END REPORT
    ;
-------------------------------------------------------------------------------------------------------------------------------
 ST SUM V  8  PG  PG   I SP  MAXIMUM TB COLUMN  DT  PG  STRT LIN SL RH .......SUBFILE..... C E A S R S N      C  SEQUENCE   LIST
 TY RPT S LPI WID HGT  M FRM LPP PGS TO HEADING POS POS PAGE NUM UB PN ENT    NAME   BLK/ FMT H F G T S U D      O  73 TO 80    SEQ
 MP     P            G           LT TYP POS        NUM     ML TD REC          BUF     C C S L T M S      N
-------------------------------------------------------------------------------------------------------------------------------
(E1)                                                                                                         (ASLGEN'D)   164
-------------------------------------------------------------------------------------------------------------------------------
         SP  Q              E N S D C S T C C M M A G D Q PCT/RAT     P              PARTIAL   SEQUENCE   LIST
STMT SEQ BF  L  FIELD  SEG- N P E E T U O U N A I V M E L  FIELD   RA- C     PICTURE    STRT NO  73 TO 80    SEQ
TYPE NO. COL F  NAME     LVL D R Q S L B T M T X N G R C F  NAME    TIO T      EDIT      CHR CHR
-------------------------------------------------------------------------------------------------------------------------------
(R1)          (CUSTNAME) 001-1                                                                               (ASLGEN'D)   165
(R1)          (CUSTNO  ) 001-1                                                                               (ASLGEN'D)   166
(R1)          (ORDERNO ) 002-2                                                                               (ASLGEN'D)   167
(R1)          (ORPERSON) 002-2                                                                               (ASLGEN'D)   168
(R1)          (ORDRDATE) 002-2                                                                               (ASLGEN'D)   169
-------------------------------------------------------------------------------------------------------------------------------
STMT  SEQ                                                                                                    SEQUENCE   LIST
TYPE  NO.                                        TEXT                                                        73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(T1) (101)                 (UNQUALIFIED FIND#                                        )                         (ASLGEN'D)   170
NOV 15, 2001  12.34.58                                                                                          PAGE     24
** MK4W204  TYPE 0  NUMBER OF MESSAGES PRINTED IS    4.                                                                  *****
                    M4OLD    -       5 RECORDS INPUT
                    M4REPO   -     242 RECORDS OUTPUT
                    M4SUBF1  -      28 RECORDS OUTPUT
                    M4INPUT  -     144 RECORDS INPUT
                    M4LIST   -     795 RECORDS OUTPUT
                      15 TRACKS ASSIGNED TO M4LIB    --   13 TRACKS NOT FULL. LIBRARY DIRECTORY BLOCKING FACTOR IS  800.
** MK4W209  TYPE 0  1007656 BYTES OF MAIN STORAGE UNUSED DURING DECODING PHASE                                       *****
                    1004432 BYTES OF MAIN STORAGE UNUSED DURING COMPILATION PHASE
                     622424 BYTES OF MAIN STORAGE UNUSED DURING PROCESSING PHASE
** MK4ED02  TYPE 0  REQUEST-RPPARALL  REPORT-1  REQUESTOR ID-                                                       *****
                                    SIMPLE REPORT USING MULTI-PATH PARALLEL LOOPING                               PAGE 1
-------------------------------------------------------------------------------------------------------------------------------
  CUSTOMER            CUSTOMER            ORDER   ORDER   SALES   INSTALLATION  OPERATING
    NUMBER              NAME              NUMBER  DATE    PERSON     NUMBER      SYSTEM
-------------------------------------------------------------------------------------------------------------------------------
   00002    FINANCIAL SERVICES             20103   033101  JJK        A031      OS/390
   00002    FINANCIAL SERVICES             20105   091501  MBK          -           -
   00003    THE INSURANCE COMPANY          30101   052301  JCR        B051      VSE/ESA
   00003    THE INSURANCE COMPANY          30102   052301  TAH        G077      OS/390
   00003    THE INSURANCE COMPANY            -       -     -        M452      VM/CMS
   00003    THE INSURANCE COMPANY            -       -     -        R802      OS/390
   00005    RECREATIONAL TOYS                -       -     -        C513      VM/CMS
   00010    HOME IMPROVEMENT SUPPLIES      01101   011301  GOA        D432      OS/390
   00010    HOME IMPROVEMENT SUPPLIES      01102   011301  TAH          -           -
   00010    HOME IMPROVEMENT SUPPLIES      01103   021501  TAH          -           -
   00010    HOME IMPROVEMENT SUPPLIES      01104   021501  JJK          -           -
   00015    CLOTHES LINE INC                 -       -     -        E919      OS/390
** MK4FT06  TYPE 1       20 FIELDS WERE PRINTED AS INVALID, MISSING OR OVERFLOW.                                    *****
** MK4FT03  TYPE 0  END OF REPORT.                                                                                  *****
** MK4ED02  TYPE 0  REQUEST-REPORT1   REPORT-1  REQUESTOR ID-                                                       *****
                                    LOOP DOMAIN LIMITED BY PROCEDURE PROCLOOP                                     PAGE 1
-------------------------------------------------------------------------------------------------------------------------------
            CUSTOMER            CUSTOMER  ORDER   ORDER   SALES
              NAME               NUMBER   NUMBER  DATE    PERSON
-------------------------------------------------------------------------------------------------------------------------------
  FINANCIAL SERVICES              00002    20103   033101  JJK
  FINANCIAL SERVICES              00002    20105   091501  MBK
  THE INSURANCE COMPANY           00003    30101   052301  JCR
  THE INSURANCE COMPANY           00003    30102   052301  TAH
  HOME IMPROVEMENT SUPPLIES       00010    01101   011301  GOA
  HOME IMPROVEMENT SUPPLIES       00010    01102   011301  TAH
  HOME IMPROVEMENT SUPPLIES       00010    01103   021501  TAH
  HOME IMPROVEMENT SUPPLIES       00010    01104   021501  JJK
** MK4FT03  TYPE 0  END OF REPORT.                                                                                  *****
** MK4ED02  TYPE 0  REQUEST-REPORT1A  REPORT-1  REQUESTOR ID-                                                       *****
                                          LOOPS INITIATED IN PROCEDURE                                        PAGE 1
                                            RANGE LIMITED IN REPORT
-------------------------------------------------------------------------------------------------------------------------------
            CUSTOMER            CUSTOMER
              NAME               NUMBER
```

Figure D-38 Multi-Path Nested Loop in the Subfile Output Command Group (Page 10 of 12)

```
------------------------------------------------------------------------------------------------------------------------
  FINANCIAL SERVICES              00002
  FINANCIAL SERVICES              00002
  THE INSURANCE COMPANY           00003
  THE INSURANCE COMPANY           00003
  HOME IMPROVEMENT SUPPLIES       00010
  HOME IMPROVEMENT SUPPLIES       00010
  HOME IMPROVEMENT SUPPLIES       00010
  HOME IMPROVEMENT SUPPLIES       00010
** MK4FT03  TYPE 0  END OF REPORT.                                                                          *****
** MK4ED02  TYPE 0  REQUEST-REPORT2   REPORT-1  REQUESTOR ID-                                              *****
                                              LOOP STARTS IN REPORT                                 PAGE 1
------------------------------------------------------------------------------------------------------------------------
           CUSTOMER            CUSTOMER  ORDER   ORDER   SALES
             NAME               NUMBER   NUMBER  DATE    PERSON
------------------------------------------------------------------------------------------------------------------------
  FINANCIAL SERVICES              00002    20103   033101   JJK
  FINANCIAL SERVICES              00002    20105   091501   MBK
  THE INSURANCE COMPANY           00003    30101   052301   JCR
  THE INSURANCE COMPANY           00003    30102   052301   TAH
  HOME IMPROVEMENT SUPPLIES       00010    01101   011301   GOA
  HOME IMPROVEMENT SUPPLIES       00010    01102   011301   TAH
  HOME IMPROVEMENT SUPPLIES       00010    01103   021501   TAH
  HOME IMPROVEMENT SUPPLIES       00010    01104   021501   JJK
** MK4FT03  TYPE 0  END OF REPORT.                                                                          *****
** MK4ED02  TYPE 0  REQUEST-REPORT4   REPORT-1  REQUESTOR ID-                                              *****
                                    SINGLE-PATH NESTED LOOP STARTED IN REPORT                       PAGE 1
                                          HIERARCHIACAL ORDER
------------------------------------------------------------------------------------------------------------------------
           CUSTOMER            CUSTOMER  ORDER   ORDER   SALES   SHIPPING   DATE
             NAME               NUMBER   NUMBER  DATE    PERSON   NUMBER   SHIPPED
------------------------------------------------------------------------------------------------------------------------
  FINANCIAL SERVICES              00002    20103   033101   JJK      D25C    042501
  FINANCIAL SERVICES              00002    20105   091501   MBK      I15J    091501
  THE INSURANCE COMPANY           00003    30101   052301   JCR      E28X    052801
  THE INSURANCE COMPANY           00003    30102   052301   TAH       -        -
  HOME IMPROVEMENT SUPPLIES       00010    01101   011301   GOA      A25G    012501
  HOME IMPROVEMENT SUPPLIES       00010    01102   011301   TAH       -        -
  HOME IMPROVEMENT SUPPLIES       00010    01103   021501   TAH       -        -
  HOME IMPROVEMENT SUPPLIES       00010    01104   021501   JJK      B19K    021901
** MK4FT06  TYPE 1        6 FIELDS WERE PRINTED AS INVALID, MISSING OR OVERFLOW.                           *****
** MK4FT03  TYPE 0  END OF REPORT.                                                                         *****
** MK4ED02  TYPE 0  REQUEST-REPORT5   REPORT-1  REQUESTOR ID-                                             *****
                                    SINGLE-PATH NESTED LOOP STARTED IN REPORT                       PAGE 1
                                          INVERSE HIERARCHIACAL ORDER
------------------------------------------------------------------------------------------------------------------------
  SHIPPING   DATE    ORDER   ORDER   SALES            CUSTOMER             CUSTOMER
    NUMBER   SHIPPED  NUMBER  DATE    PERSON             NAME               NUMBER
------------------------------------------------------------------------------------------------------------------------
     D25C    042501   20103   033101   JJK    FINANCIAL SERVICES             00002
     I15J    091501   20105   091501   MBK    FINANCIAL SERVICES             00002
     E28X    052801   30101   052301   JCR    THE INSURANCE COMPANY          00003
     A25G    012501   01101   011301   GOA    HOME IMPROVEMENT SUPPLIES      00010
     B19K    021901   01104   021501   JJK    HOME IMPROVEMENT SUPPLIES      00010
** MK4FT03  TYPE 0  END OF REPORT.                                                                          *****
** MK4ED02  TYPE 0  REQUEST-REPORT6   REPORT-1  REQUESTOR ID-                                              *****
                                          MULTI-PATH NESTED LOOPING                                 PAGE 1
------------------------------------------------------------------------------------------------------------------------
           CUSTOMER            CUSTOMER  ORDER   SALES   INSTALLATION  OPERATING
             NAME               NUMBER   NUMBER  PERSON     NUMBER      SYSTEM
------------------------------------------------------------------------------------------------------------------------
  FINANCIAL SERVICES              00002    20103   JJK       A031       OS/390
                                           20105   MBK       A031       OS/390
  HOME IMPROVEMENT SUPPLIES       00010    01101   GOA       D432       OS/390
                                           01102   TAH       D432       OS/390
                                           01103   TAH       D432       OS/390
                                           01104   JJK       D432       OS/390
  THE INSURANCE COMPANY           00003    30101   JCR       B051       VSE/ESA
                                           30101   JCR       G077       OS/390
                                           30101   JCR       M452       VM/CMS
                                           30101   JCR       R802       OS/390
                                           30102   TAH       B051       VSE/ESA
                                           30102   TAH       G077       OS/390
                                           30102   TAH       M452       VM/CMS
                                           30102   TAH       R802       OS/390
** MK4FT03  TYPE 0  END OF REPORT.                                                                          *****
** MK4ED02  TYPE 0  REQUEST-REPORT7   REPORT-1  REQUESTOR ID-                                              *****
                                          MULTI-PATH PARALLEL LOOPING                               PAGE 1
------------------------------------------------------------------------------------------------------------------------
           CUSTOMER            CUSTOMER  ORDER   SALES   INSTALLATION  OPERATING
             NAME               NUMBER   NUMBER  PERSON     NUMBER      SYSTEM
------------------------------------------------------------------------------------------------------------------------
  FINANCIAL SERVICES              00002    20103   JJK       A031       OS/390
                                           20105   MBK         -          -
  HOME IMPROVEMENT SUPPLIES       00010    01101   GOA       D432       OS/390
```

Figure D-38 Multi-Path Nested Loop in the Subfile Output Command Group (Page 11 of 12)

```
                                        01102    TAH          -            -
                                        01103    TAH          -            -
                                        01104    JJK          -            -
  THE INSURANCE COMPANY          00003  30101    JCR        B051      VSE/ESA
                                        30102    TAH        G077      OS/390
                                            -      -        M452      VM/CMS
                                            -      -        R802      OS/390
** MK4FT06  TYPE 1       12 FIELDS WERE PRINTED AS INVALID, MISSING OR OVERFLOW.                                    *****
** MK4FT03  TYPE 0  END OF REPORT.                                                                                   *****
** MK4ED02  TYPE 0  REQUEST-REPORT8   REPORT-1  REQUESTOR ID-                                                        *****
                                                MULTI-PATH PARALLEL LOOPING                                  PAGE 1
-----------------------------------------------------------------------------------------------------------------------------
            CUSTOMER            CUSTOMER  ORDER   SALES   INSTALLATION  OPERATING
              NAME               NUMBER   NUMBER  PERSON     NUMBER      SYSTEM
-----------------------------------------------------------------------------------------------------------------------------
  FINANCIAL SERVICES             00002   20103    JJK        A031      OS/390
                                         20105    MBK        A031      OS/390
  HOME IMPROVEMENT SUPPLIES      00010   01101    GOA        D432      OS/390
                                         01102    TAH        D432      OS/390
                                         01103    TAH        D432      OS/390
                                         01104    JJK        D432      OS/390
  THE INSURANCE COMPANY          00003   30101    JCR        B051      VSE/ESA
                                         30101    JCR        G077      OS/390
                                         30101    JCR        M452      VM/CMS
                                         30101    JCR        R802      OS/390
                                         30102    TAH        B051      VSE/ESA
                                         30102    TAH        G077      OS/390
                                         30102    TAH        M452      VM/CMS
                                         30102    TAH        R802      OS/390
** MK4FT03  TYPE 0  END OF REPORT.                                                                                   *****
** MK4ED02  TYPE 0  REQUEST-SELECT    REPORT-1  REQUESTOR ID-                                                        *****
                                             LOOPING LIMITED BY SELECTION CONTROL                            PAGE 1
-----------------------------------------------------------------------------------------------------------------------------
            CUSTOMER            CUSTOMER  ORDER   SALES   INSTALLATION  OPERATING
              NAME               NUMBER   NUMBER  PERSON     NUMBER      SYSTEM
-----------------------------------------------------------------------------------------------------------------------------
  CLOTHES LINE INC               00015       -      -        E919      OS/390
  FINANCIAL SERVICES             00002   20103    JJK        A031      OS/390
  HOME IMPROVEMENT SUPPLIES      00010   01101    GOA        D432      OS/390
  RECREATIONAL TOYS              00005       -      -        C513      VM/CMS
  THE INSURANCE COMPANY          00003   30101    JCR        B051      VSE/ESA
** MK4FT06  TYPE 1        4 FIELDS WERE PRINTED AS INVALID, MISSING OR OVERFLOW.                                    *****
** MK4FT03  TYPE 0  END OF REPORT.                                                                                   *****
** MK4ED02  TYPE 0  REQUEST-REPORT9   REPORT-1  REQUESTOR ID-                                                        *****
                                                     FIND ORDER 01104                                        PAGE 1
-----------------------------------------------------------------------------------------------------------------------------
            CUSTOMER            CUSTOMER  ORDER   SALES   ORDER
              NAME               NUMBER   NUMBER  PERSON  DATE
-----------------------------------------------------------------------------------------------------------------------------
  HOME IMPROVEMENT SUPPLIES      00010   01104    JJK    021501
** MK4FT03  TYPE 0  END OF REPORT.                                                                                   *****
** MK4ED02  TYPE 0  REQUEST-REP10     REPORT-1  REQUESTOR ID-                                                        *****
                                                     UNQUALIFIED FIND                                        PAGE 1
-----------------------------------------------------------------------------------------------------------------------------
            CUSTOMER            CUSTOMER  ORDER   SALES   ORDER
              NAME               NUMBER   NUMBER  PERSON  DATE
-----------------------------------------------------------------------------------------------------------------------------
  FINANCIAL SERVICES             00002   20103    JJK    033101
  THE INSURANCE COMPANY          00003   30101    JCR    052301
  HOME IMPROVEMENT SUPPLIES      00010   01101    GOA    011301
** MK4FT03  TYPE 0  END OF REPORT.                                                                                   *****
** MK4W204  TYPE 0  NUMBER OF MESSAGES PRINTED IS   28.                                                              *****
                   M4REPO   -      241 RECORDS INPUT
                   M4LIST   -      189 RECORDS OUTPUT
** MK4W209  TYPE 0  1050192 BYTES OF MAIN STORAGE UNUSED DURING REPORTING PHASE                                      *****
```

Figure D-38 Multi-Path Nested Loop in the Subfile Output Command Group (Page 12 of 12)

# Reporting

## Inventory Check List Report

```
--------------------------------------------------------------------------------------------------------------------------------
   RUN     STMT    FILE    O N S D U T R A B S D S L  RB  R S M R S          START          END            SEQUENCE   LIST
   NAME    TYPE    NAME    L E E I P R E U U R L C S  PL  J O O F S          SEARCH         SEARCH         73 TO 80    SEQ
                           D W Q R D N P D F T M N T  TK  T P P O R
--------------------------------------------------------------------------------------------------------------------------------
 (ASLRC   ) (RC) (ITEMFILE) (S)         (S)          (G)          (R)                                    (ASLGEN'D)     1
0     CONTROL  SYSDATE 061402   LISTGEN
      FILE MASTER INPUT, NAME ITEMFILE
      FILE REPORT
      ;
--------------------------------------------------------------------------------------------------------------------------------
   RUN     STMT  LOGICAL  LBL  DTF/DD     KEY      KEY      KEY     QUALIFIER   USER I/O  ARRAY      PASSWORD/  SEQUENCE   LIST
   NAME    TYPE  FILENAME       NAME     FIELD-1  FIELD-2  FIELD-3  ICF CRD DBI   MODULE   QUAL       AUTH.-ID   73 TO 80    SEQ
--------------------------------------------------------------------------------------------------------------------------------
 (ASLRC   ) (RF) (ITEMFILE) (S) (M4OLD    )                                                              (ASLGEN'D)     2
--------------------------------------------------------------------------------------------------------------------------------
   RUN     STMT  PARAMETER                                                                              SEQUENCE   LIST
   NAME    TYPE    NAME                          PARAMETER VALUE                                        73 TO 80    SEQ
--------------------------------------------------------------------------------------------------------------------------------
 (ASLRC   ) (RP) (SYSDATE )        (061402                                    )                        (ASLGEN'D)     3


================================================================================================================================
=                                                                                                                              =
=                                              F I L E    S U M M A R Y                                                        =
=                                                                                                                              =
= JOBNAME: HANME02K                          STEPNAME: F9                                                                      =
=                                                                                                                              =
=  FILE NAME: M4OLD                          DSNAME: BUILDER.R140.ITEM.FILE.R09#1                                              =
=  FILE NAME: M4LIB                          DSNAME: BUILDER.R140.M4LIB.R09#1                                                  =
=                                                                                                                              =
================================================================================================================================
1NOV 27, 2001  13.06.56                                                                                          PAGE      3
                                            ******************************
                                            *  PROC NAME - TEMP          *
                                            *  INPUT STREAM PROCEDURE    *
                                            ******************************
--------------------------------------------------------------------------------------------------------------------------------
 STMT REPORT                              MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB       SEQUENCE   LIST
 TYPE  DATE          REQUESTOR ID        ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM       73 TO 80    SEQ
--------------------------------------------------------------------------------------------------------------------------------
 (ER)                                                                          (N)                                (ASLGEN'D)     4
0     TEMP: PROC ; define temp fields
      COSTVALU: FIELD Z, LEN 7, DEC 2, HEAD 'VALUE' 'AT COST',      FLT '$'
      SALEVALU: FIELD Z, LEN 7, DEC 2, HEAD 'VALUE AT' 'SALE PRICE', FLT '$'
      NEWCOST:  FIELD Z, LEN 7, DEC 2, HEAD 'EXPECTED' 'NEW COST', FLT '$'
      NEWPRICE: FIELD Z, LEN 7, DEC 2, HEAD 'RECOMMENDED' 'NEW PRICE', FLT '$'
      DIFCOST:  FIELD Z, LEN 7, DEC 2, HEAD 'ESTIMATED' 'COST VALUE'
      DIFPRICE: FIELD Z, LEN 7, DEC 2, HEAD 'ESTIMATED' 'PRICE VALUE'
      END PROC
      ;
--------------------------------------------------------------------------------------------------------------------------------
 STMT   FIELD    FIELD FLD DEC OUTPT EDIT     INITIAL        LINE 1 OF       LINE 2 OF                          SEQUENCE   LIST
 TYPE   NAME     LNGTH TYP PLC EDIT  LGTH      VALUE        COLUMN HEADING  COLUMN HEADING                      73 TO 80    SEQ
--------------------------------------------------------------------------------------------------------------------------------
 (TF) (COSTVALU) (  7) (Z) (2) ($  )                    (VALUE        ) (AT COST      )                      (ASLGEN'D)     5
 (TF) (SALEVALU) (  7) (Z) (2) ($  )                    (VALUE AT     ) (SALE PRICE   )                      (ASLGEN'D)     6
 (TF) (NEWCOST ) (  7) (Z) (2) ($  )                    (EXPECTED     ) (NEW COST     )                      (ASLGEN'D)     7
 (TF) (NEWPRICE) (  7) (Z) (2) ($  )                    (RECOMMENDED  ) (NEW PRICE    )                      (ASLGEN'D)     8
 (TF) (DIFCOST ) (  7) (Z) (2)                          (ESTIMATED    ) (COST VALUE   )                      (ASLGEN'D)     9
 (TF) (DIFPRICE) (  7) (Z) (2)                          (ESTIMATED    ) (PRICE VALUE  )                      (ASLGEN'D)    10
 ** MK4CT06  TYPE 1  REQUEST TEMP     CONTAINS COMMENTS AND TEMPORARY FIELDS ONLY.                                      *****
1NOV 27, 2001  13.06.56                                                                                          PAGE      4
                                            ******************************
                                            *  PROC NAME - INVCHK1       *
                                            *  INPUT STREAM PROCEDURE    *
                                            ******************************
--------------------------------------------------------------------------------------------------------------------------------
 STMT REPORT                              MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB       SEQUENCE   LIST
 TYPE  DATE          REQUESTOR ID        ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM       73 TO 80    SEQ
--------------------------------------------------------------------------------------------------------------------------------
 (ER) (TODAY )                                                                 (N)                                (ASLGEN'D)    11
0     INVCHK1: REPORT,
            ITEMS ITEMNO, ITEMNAME, QANONHND, REORDPT, ORDQTY,
                  UNITCOST, ITMPRICE, VENDOR, ORDERNO, QTYBKORD
            FORMAT DATEFMT TODAY
```

Figure D-39 Inventory Check List Report (Page 1 of 13)

```
            TITLE 'INVENTORY CHECKLIST'
    END REPORT
            ; customize only date format
    ;
-------------------------------------------------------------------------------------------------------------------------------
 ST SUM V  8  PG  PG   I SP  MAXIMUM TB COLUMN  DT  PG  STRT LIN SL RH .......SUBFILE........ C E A S R S N      C  SEQUENCE   LIST
 TY RPT S LPI WID HGT  M FRM LPP PGS TO HEADING POS POS PAGE NUM UB PN ENT    NAME   BLK/ FMT H F G T S U D      O  73 TO 80    SEQ
 MP     P             G           LT TYP POS         NUM      ML TD REC           BUF      C C S L T M S      N
-------------------------------------------------------------------------------------------------------------------------------
(E1)                                                                                                          (ASLGEN'D)    12
-------------------------------------------------------------------------------------------------------------------------------
         SP  Q              E N S D C S T C C M M A G D Q PCT/RAT     P              PARTIAL   SEQUENCE   LIST
STMT SEQ BF  L  FIELD  SEG-  N P E E T U O U N A I V M E L FIELD   RA- C     PICTURE     STRT NO  73 TO 80   SEQ
TYPE NO. COL F  NAME     LVL D R Q S L B T M T X N G R C F NAME    TIO T      EDIT       CHR CHR
-------------------------------------------------------------------------------------------------------------------------------
(R1)          (ITEMNO  ) 001-1                                                                                (ASLGEN'D)    13
(R1)          (ITEMNAME) 001-1                                                                                (ASLGEN'D)    14
(R1)          (QANONHND) 001-1                                                                                (ASLGEN'D)    15
(R1)          (REORDPT ) 001-1                                                                                (ASLGEN'D)    16
(R1)          (ORDQTY  ) 001-1                                                                                (ASLGEN'D)    17
(R1)          (UNITCOST) 001-1                                                                                (ASLGEN'D)    18
(R1)          (ITMPRICE) 001-1                                                                                (ASLGEN'D)    19
(R1)          (VENDOR  ) 001-1                                                                                (ASLGEN'D)    20
(R1)          (ORDERNO ) 002-2                                                                                (ASLGEN'D)    21
(R1)          (QTYBKORD) 002-2                                                                                (ASLGEN'D)    22
-------------------------------------------------------------------------------------------------------------------------------
STMT  SEQ                                                                                                     SEQUENCE   LIST
TYPE  NO.                                          TEXT                                                       73 TO 80   SEQ
-------------------------------------------------------------------------------------------------------------------------------
 (T1) (101)                (INVENTORY CHECKLIST#                                  )                           (ASLGEN'D)    23
1NOV 27, 2001  13.06.56                                                                                          PAGE     5
                                          ****************************
                                          *  PROC NAME - ORDERPT     *
                                          *  INPUT STREAM PROCEDURE  *
                                          ****************************
-------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                            MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB      SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID      ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM      73 TO 80   SEQ
-------------------------------------------------------------------------------------------------------------------------------
 (ER) (DATE  )                                                          (N)                                    (ASLGEN'D)    24
0     ORDERPT: REPORT,
            ITEMS ITEMNO, ITEMNAME, QANONHND, REORDPT, ORDQTY
              FORMAT DATEFMT DATE
              ITEM QANONHND SPACES 10
              ITEM ORDQTY   SPACES 10
            TITLE 'ORDERING REPORT'
    END REPORT
            ; customize column spacing and date format
    ;
-------------------------------------------------------------------------------------------------------------------------------
 ST SUM V  8  PG  PG   I SP  MAXIMUM TB COLUMN  DT  PG  STRT LIN SL RH .......SUBFILE........ C E A S R S N      C  SEQUENCE   LIST
 TY RPT S LPI WID HGT  M FRM LPP PGS TO HEADING POS POS PAGE NUM UB PN ENT    NAME   BLK/ FMT H F G T S U D      O  73 TO 80   SEQ
 MP     P             G           LT TYP POS         NUM      ML TD REC           BUF      C C S L T M S      N
-------------------------------------------------------------------------------------------------------------------------------
(E1)                                                                                                          (ASLGEN'D)    25
-------------------------------------------------------------------------------------------------------------------------------
         SP  Q              E N S D C S T C C M M A G D Q PCT/RAT     P              PARTIAL   SEQUENCE   LIST
STMT SEQ BF  L  FIELD  SEG-  N P E E T U O U N A I V M E L FIELD   RA- C     PICTURE     STRT NO  73 TO 80   SEQ
TYPE NO. COL F  NAME     LVL D R Q S L B T M T X N G R C F NAME    TIO T      EDIT       CHR CHR
-------------------------------------------------------------------------------------------------------------------------------
(R1)          (ITEMNO  ) 001-1                                                                                (ASLGEN'D)    26
(R1)          (ITEMNAME) 001-1                                                                                (ASLGEN'D)    27
(R1)     (10) (QANONHND) 001-1                                                                                (ASLGEN'D)    28
(R1)          (REORDPT ) 001-1                                                                                (ASLGEN'D)    29
(R1)     (10) (ORDQTY  ) 001-1                                                                                (ASLGEN'D)    30
-------------------------------------------------------------------------------------------------------------------------------
STMT  SEQ                                                                                                     SEQUENCE   LIST
TYPE  NO.                                          TEXT                                                       73 TO 80   SEQ
-------------------------------------------------------------------------------------------------------------------------------
 (T1) (101)                (ORDERING REPORT#                                      )                           (ASLGEN'D)    31
1NOV 27, 2001  13.06.56                                                                                          PAGE     6
                                          ****************************
                                          *  PROC NAME - INVCOUNT    *
                                          *  INPUT STREAM PROCEDURE  *
                                          ****************************
-------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                            MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB      SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID      ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM      73 TO 80   SEQ
-------------------------------------------------------------------------------------------------------------------------------
 (ER)                                                                   (N)                                    (ASLGEN'D)    32
0     INVCOUNT: PROC
       INSTOCK: FIELD TYPE C, LENGTH 5, INIT '____', HEADING 'IN' 'STOCK'
       REPORT ITEMS ITEMNO, PF(ITEMNAME,1,15), T.INSTOCK, REORDPT, ORDQTY
              FORMAT BORDER NO, DETAILSPACING 2,
                     IMAGES 2,   WIDTH 55,   HEIGHT 35
              TITLE 'INVENTORY COUNT FORM'
```

Figure D-39 Inventory Check List Report (Page 2 of 13)

```
     END REPORT
     END PROC
             ; customize format
     ;
-------------------------------------------------------------------------------------------------------------------------------
STMT   FIELD    FIELD FLD DEC OUTPT EDIT      INITIAL          LINE 1 OF       LINE 2 OF                                SEQUENCE   LIST
TYPE   NAME     LNGTH TYP PLC EDIT  LGTH       VALUE         COLUMN HEADING  COLUMN HEADING                            73 TO 80   SEQ
-------------------------------------------------------------------------------------------------------------------------------
(TF) (INSTOCK ) (  5) (C)              (_____          ) (IN           ) (STOCK        )                          (ASLGEN'D)    33
-------------------------------------------------------------------------------------------------------------------------------
 ST SUM V  8  PG  PG   I SP  MAXIMUM TB COLUMN  DT  PG  STRT LIN SL RH .......SUBFILE..... C E A S R S N      C  SEQUENCE   LIST
 TY RPT S LPI WID HGT  M FRM LPP PGS TO HEADING POS POS PAGE NUM UB PN ENT    NAME   BLK/ FMT H F G T S U D      O  73 TO 80   SEQ
 MP     P            G            LT TYP POS         NUM      ML TD REC          BUF      C C S L T M S      N
-------------------------------------------------------------------------------------------------------------------------------
(E1)    (2)    (055,035) (2)                                                          (N)                      (ASLGEN'D)    34
-------------------------------------------------------------------------------------------------------------------------------
         SP  Q              E N  S  D  C  S  T  C  C  M  M  A  G  D Q PCT/RAT     P                PARTIAL   SEQUENCE   LIST
STMT SEQ  BF  L  FIELD   SEG-  N P  E  E  T  U  O  U  N  A  I  V  M  E L  FIELD   RA- C     PICTURE     STRT NO   73 TO 80   SEQ
TYPE NO.  COL F  NAME     LVL  D R  Q  S  L  B  T  M  T  X  N  G  R  C F  NAME    TIO T      EDIT       CHR CHR
-------------------------------------------------------------------------------------------------------------------------------
(R1)          (ITEMNO  ) 001-1                                                                                (ASLGEN'D)    35
(R1)          (ITEMNAME) 001-1                                                                 ( 1) (15)      (ASLGEN'D)    36
(R1)        (T,INSTOCK )                                                                                      (ASLGEN'D)    37
(R1)          (REORDPT ) 001-1                                                                                (ASLGEN'D)    38
(R1)          (ORDQTY  ) 001-1                                                                                (ASLGEN'D)    39
-------------------------------------------------------------------------------------------------------------------------------
STMT  SEQ                                                                                                     SEQUENCE   LIST
TYPE  NO.                                        TEXT                                                         73 TO 80   SEQ
-------------------------------------------------------------------------------------------------------------------------------
(T1) (101)                (INVENTORY COUNT FORM#                                 )                            (ASLGEN'D)    40
1NOV 27, 2001  13.06.56                                                                                          PAGE     7
                                                  ****************************
                                                  *  PROC NAME - SORTBRK     *
                                                  *  INPUT STREAM PROCEDURE  *
                                                  ****************************
-------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                              MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID        ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM       73 TO 80   SEQ
-------------------------------------------------------------------------------------------------------------------------------
(ER)                                                                             (N)                          (ASLGEN'D)    41
0     SORTBRK: PROC
      LET T.COSTVALU = QANONHND * UNITCOST
      LET T.SALEVALU = QANONHND * ITMPRICE
       REPORT  ITEMNO, ITEMNAME, VENDOR, QANONHND, ORDQTY,
            UNITCOST, T.COSTVALU, ITMPRICE, T.SALEVALU
         FORMAT DATEFMT DATE
         TITLE 'STOCK COST AND PRICE VALUE COMPARISON'
         ORDER BY VENDOR
         GROUP BY VENDOR
       END REPORT
      END PROC
             ; introduce ordering and grouping
     ;
-------------------------------------------------------------------------------------------------------------------------------
STMT SEQ  LOG CON .....OPERAND-A.....  OPER  .............OPERAND-B.............  ......RESULT......  PARTIAL FIELD  SEQUENCE   LIST
TYPE NO.  LEV CTR QLF FIELD   SEG-LVL  ATN   QLF FIELD   SEG-LVL  (OR CONSTANT)   QLF FIELD  SEG-LVL  STRT NUMB OPR  73 TO 80   SEQ
-------------------------------------------------------------------------------------------------------------------------------
(PR) (100)          (QANONHND) 001-1  (* )     (UNITCOST) 001-1                 (T,COSTVALU)                      (ASLGEN'D)    42
(PR) (101)          (QANONHND) 001-1  (* )     (ITMPRICE) 001-1                 (T,SALEVALU)                      (ASLGEN'D)    43
(PR) (102)                            (GO)     (SUB REPT___6                 )                                  (ASLGEN'D)    44
1NOV 27, 2001  13.06.56                                                                                          PAGE     8
                                                  ****************************
                                                  *  REQUEST NAME - REPT___6 *
                                                  *  GENERATED REQUEST       *
                                                  ****************************
-------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                              MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID        ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM       73 TO 80   SEQ
-------------------------------------------------------------------------------------------------------------------------------
(ER) (DATE  )                                                                    (S)                          (ASLGEN'D)    45
-------------------------------------------------------------------------------------------------------------------------------
 ST SUM V  8  PG  PG   I SP  MAXIMUM TB COLUMN  DT  PG  STRT LIN SL RH .......SUBFILE..... C E A S R S N      C  SEQUENCE   LIST
 TY RPT S LPI WID HGT  M FRM LPP PGS TO HEADING POS POS PAGE NUM UB PN ENT    NAME   BLK/ FMT H F G T S U D      O  73 TO 80   SEQ
 MP     P            G            LT TYP POS         NUM      ML TD REC          BUF      C C S L T M S      N
-------------------------------------------------------------------------------------------------------------------------------
(E1)                                                                                                          (ASLGEN'D)    46
-------------------------------------------------------------------------------------------------------------------------------
         SP  Q              E N  S  D  C  S  T  C  C  M  M  A  G  D Q PCT/RAT     P                PARTIAL   SEQUENCE   LIST
STMT SEQ  BF  L  FIELD   SEG-  N P  E  E  T  U  O  U  N  A  I  V  M  E L  FIELD   RA- C     PICTURE     STRT NO   73 TO 80   SEQ
TYPE NO.  COL F  NAME     LVL  D R  Q  S  L  B  T  M  T  X  N  G  R  C F  NAME    TIO T      EDIT       CHR CHR
-------------------------------------------------------------------------------------------------------------------------------
(R1)          (ITEMNO  ) 001-1                                                                                (ASLGEN'D)    47
(R1)          (ITEMNAME) 001-1                                                                                (ASLGEN'D)    48
(R1)          (VENDOR  ) 001-1     (1)    (1)                                                                 (ASLGEN'D)    49
(R1)          (QANONHND) 001-1                                                                                (ASLGEN'D)    50
(R1)          (ORDQTY  ) 001-1                                                                                (ASLGEN'D)    51
```

Figure D-39 Inventory Check List Report (Page 3 of 13)

```
(R1)          (UNITCOST) 001-1                                                                                    (ASLGEN'D)    52
(R1)        (T,COSTVALU)                                                                                          (ASLGEN'D)    53
(R1)          (ITMPRICE) 001-1                                                                                    (ASLGEN'D)    54
(R1)        (T,SALEVALU)                                                                                          (ASLGEN'D)    55
-------------------------------------------------------------------------------------------------------------------------------------
STMT  SEQ                                                                                                         SEQUENCE   LIST
TYPE  NO.                                          TEXT                                                           73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------------
 (T1) (101)                (STOCK COST AND PRICE VALUE COMPARISON#                  )                            (ASLGEN'D)    56
1NOV 27, 2001  13.06.56                                                                                               PAGE     9
                                          ****************************
                                          *  PROC NAME - BKORDER     *
                                          *  INPUT STREAM PROCEDURE  *
                                          ****************************
-------------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                              MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID        ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM      73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------------
 (ER)                                                                          (N)                               (ASLGEN'D)    57
0     BKORDER: PROC
       IF QTYBKORD GT 0    THEN
          REPORT VENDOR ITEMNO ITEMNAME ORDERNO QTYBKORD
              FORMAT DATEFMT DATE
              TITLE 'BACKORDERS BY VENDOR AND ITEM'
            ORDER BY VENDOR ITEMNO
            GROUP BY VENDOR
            GROUP BY ITEMNO
            TOTAL QTYBKORD BY ITEMNO
          END REPORT
       END
      END PROC
             ; multi-level control break
      ;
-------------------------------------------------------------------------------------------------------------------------------------
STMT SEQ  LOG CON .....OPERAND-A.....  OPER  .............OPERAND-B.............   ......RESULT......  PARTIAL FIELD  SEQUENCE   LIST
TYPE NO.  LEV CTR QLF FIELD   SEG-LVL  ATN   QLF FIELD   SEG-LVL  (OR CONSTANT)   QLF FIELD  SEG-LVL  STRT NUMB OPR  73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------------
(PR) (100)           (QTYBKORD) 002-2  (GT)    (D,0                          )                                    (ASLGEN'D)    58
(PR) (101)                             (NS)      (END                        )                                    (ASLGEN'D)    59
(PR) (102)                             (GO)      (SUB REPT___7               )                                    (ASLGEN'D)    60
1NOV 27, 2001  13.06.56                                                                                               PAGE     10
                                          ****************************
                                          *  REQUEST NAME - REPT___7 *
                                          *  GENERATED REQUEST       *
                                          ****************************
-------------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                              MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID        ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM      73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------------
(ER) (DATE  )                                                                 (S)                               (ASLGEN'D)    61
-------------------------------------------------------------------------------------------------------------------------------------
 ST SUM V  8  PG  PG   I SP  MAXIMUM TB COLUMN  DT  PG  STRT LIN SL RH .......SUBFILE........ C E A S R S N      C  SEQUENCE   LIST
 TY RPT S LPI WID HGT  M FRM LPP PGS TO HEADING POS POS PAGE NUM UB PN ENT    NAME   BLK/ FMT H F G T S U D      O  73 TO 80    SEQ
 MP     P             G            LT TYP POS         NUM      ML TD REC          BUF      C C S L T M S      N
-------------------------------------------------------------------------------------------------------------------------------------
(E1)                                                                                                              (ASLGEN'D)    62
-------------------------------------------------------------------------------------------------------------------------------------
         SP  Q              E  N  S  D  C  S  T  C  C  M  M  A  G  D Q PCT/RAT     P                PARTIAL   SEQUENCE   LIST
STMT SEQ BF  L  FIELD  SEG-  N  P  E  E  T  U  O  U  N  A  I  V  M  E L  FIELD   RA- C      PICTURE    STRT NO  73 TO 80    SEQ
TYPE NO. COL F  NAME     LVL  D  R  Q  S  L  B  T  M  T  X  N  G  R  C F  NAME    TIO T       EDIT      CHR CHR
-------------------------------------------------------------------------------------------------------------------------------------
(R1)          (VENDOR  ) 001-1      (1)   (1)                                                                   (ASLGEN'D)    63
(R1)          (ITEMNO  ) 001-1      (2)   (2)                                                                   (ASLGEN'D)    64
(R1)          (ITEMNAME) 001-1                                                                                  (ASLGEN'D)    65
(R1)          (ORDERNO ) 002-2                                                                                  (ASLGEN'D)    66
(R1)          (QTYBKORD) 002-2                 (2)                                                              (ASLGEN'D)    67
-------------------------------------------------------------------------------------------------------------------------------------
STMT  SEQ                                                                                                         SEQUENCE   LIST
TYPE  NO.                                          TEXT                                                           73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------------
 (T1) (101)                (BACKORDERS BY VENDOR AND ITEM#                         )                            (ASLGEN'D)    68
1NOV 27, 2001  13.06.56                                                                                               PAGE     11
                                          ****************************
                                          *  PROC NAME - SUMBRK      *
                                          *  INPUT STREAM PROCEDURE  *
                                          ****************************
-------------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                              MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID        ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM      73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------------
 (ER)                                                                          (N)                               (ASLGEN'D)    69
0     SUMBRK: PROC
      LET T.COSTVALU = QANONHND * UNITCOST
      LET T.SALEVALU = QANONHND * ITMPRICE
      LET T.PRODCODE = PF(ITEMNO,1,2)
       ;
```

Figure D-39 Inventory Check List Report (Page 4 of 13)

```
      REPORT ITEMS  VENDOR, T.PRODCODE, ITEMNO, ITEMNAME, QANONHND,
                    ORDQTY, UNITCOST, T.COSTVALU, ITMPRICE, T.SALEVALU
         FORMAT DATEFMT DATE
         TITLE 'STOCK COST AND PRICE VALUE COMPARISON'
         ORDER BY VENDOR
         GROUP BY VENDOR
           TOTAL T.COSTVALU BY VENDOR
           TOTAL T.SALEVALU BY VENDOR
      END REPORT
      ; introduce summaries
      ; f261
      REPORT GRANDSUMS,
             ITEMS  VENDOR, T.PRODCODE, ITEMNO, ITEMNAME, QANONHND,
                    ORDQTY, UNITCOST, T.COSTVALU, ITMPRICE, T.SALEVALU
         FORMAT DATEFMT DATE
         TITLE 'STOCK COST AND PRICE VALUE COMPARISON'
         ORDER BY VENDOR
         GROUP BY VENDOR T.PRODCODE SUBTITLE NEWPAGE
         ITEM  VENDOR ENDLINE
           TOTAL T.COSTVALU BY VENDOR
           TOTAL T.SALEVALU BY VENDOR
      END REPORT
             ; customize summary output with subtitles
      ;f263
      REPORT SUMMARY ONLY SINGLESPACE,
             ITEMS  VENDOR, T.PRODCODE, T.COSTVALU, T.SALEVALU
         FORMAT DATEFMT DATE
         TITLE 'SUMMARY OF COST AND PRICE VALUE COMPARISON'
         ORDER BY VENDOR
         GROUP BY VENDOR T.PRODCODE
           TOTAL    T.COSTVALU BY VENDOR
           CUMULATE T.COSTVALU AT G
           TOTAL    T.SALEVALU BY VENDOR
           CUMULATE T.SALEVALU AT G
      END REPORT
             ; summary only with labels
      ;f266
      REPORT SUMMARY ONLY SINGLESPACE,
             ITEMS  VENDOR, T.PRODCODE, T.COSTVALU, T.SALEVALU
         FORMAT DATEFMT DATE SUMMARYLABELS SUPPRESS
         TITLE 'SUMMARY OF COST AND PRICE VALUE COMPARISON'
         ORDER BY VENDOR
         GROUP BY VENDOR T.PRODCODE
           TOTAL    T.COSTVALU BY VENDOR
1NOV 27, 2001  13.06.56                                                                                    PAGE     12
           CUMULATE T.COSTVALU AT G
           TOTAL    T.SALEVALU BY VENDOR
           CUMULATE T.SALEVALU AT G
      END REPORT
             ; summary only with labels suppressed
             ; f269
    END PROC
    ;
```

```
STMT SEQ  LOG CON .....OPERAND-A.....  OPER  .............OPERAND-B.............   ......RESULT......  PARTIAL FIELD  SEQUENCE   LIST
TYPE NO.  LEV CTR QLF FIELD   SEG-LVL  ATN   QLF FIELD   SEG-LVL  (OR CONSTANT)   QLF FIELD  SEG-LVL  STRT NUMB OPR  73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------------
(PR) (100)           (QANONHND) 001-1  (* )     (UNITCOST) 001-1                 (T,COSTVALU)                       (ASLGEN'D)     70
(PR) (101)           (QANONHND) 001-1  (* )     (ITMPRICE) 001-1                 (T,SALEVALU)                       (ASLGEN'D)     71
(PR) (102)                             (R )     (ITEMNO  ) 001-1                 (T,PRODCODE)         ( 1) ( 2) (B) (ASLGEN'D)     72
** MK4DI02  TYPE 0  DEFINITION OF TEMPORARY FIELD PRODCODE - FORMAT C  LENGTH   2  DECIMALS 0 EDIT LENGTH   2 .                      *****
(PR) (103)                             (GO)     (SUB REPT_12               )                                       (ASLGEN'D)     73
(PR) (104)                             (GO)     (SUB REPT__13              )                                       (ASLGEN'D)     74
(PR) (105)                             (GO)     (SUB REPT__14              )                                       (ASLGEN'D)     75
(PR) (106)                             (GO)     (SUB REPT__15              )                                       (ASLGEN'D)     76
1NOV 27, 2001  13.06.56                                                                                        PAGE     13
                                                    ****************************
                                                    *  REQUEST NAME - REPT__12  *
                                                    *  GENERATED REQUEST   __   *
                                                    ****************************
```

```
STMT REPORT                                MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID          ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM       73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------------
(ER) (DATE  )                                                                    (S)                                (ASLGEN'D)     77
-------------------------------------------------------------------------------------------------------------------------------------
 ST SUM V  8  PG  PG   I SP  MAXIMUM TB COLUMN  DT  PG  STRT LIN SL RH .......SUBFILE........ C E A S R S N      C  SEQUENCE   LIST
 TY RPT S LPI WID HGT  M FRM LPP PGS TO HEADING POS POS PAGE NUM UB PN ENT    NAME    BLK/ FMT H F G T S U D      O  73 TO 80    SEQ
 MP     P            G            LT TYP POS         NUM      ML TD REC           BUF      C C S L T M S      N
-------------------------------------------------------------------------------------------------------------------------------------
(E1)                                                                                                             (ASLGEN'D)     78
-------------------------------------------------------------------------------------------------------------------------------------
         SP  Q              E N  S  D  C  S  T  C  C  M  M  A  G  D Q PCT/RAT     P                PARTIAL   SEQUENCE   LIST
STMT SEQ  BF  L  FIELD   SEG-  N P  E  E  T  U  O  U  N  A  I  V  M  E L  FIELD   RA- C     PICTURE    STRT NO  73 TO 80    SEQ
TYPE NO.  COL F  NAME      LVL  D R  Q  S  L  B  T  M  T  X  N  G  R  C F  NAME    TIO T      EDIT      CHR CHR
-------------------------------------------------------------------------------------------------------------------------------------
```

Figure D-39 Inventory Check List Report (Page 5 of 13)

```
 (R1)          (VENDOR  ) 001-1      (1)   (1)                                                                      (ASLGEN'D)    79
 (R1)         (T,PRODCODE)                                                                                          (ASLGEN'D)    80
 (R1)          (ITEMNO  ) 001-1                                                                                     (ASLGEN'D)    81
 (R1)          (ITEMNAME) 001-1                                                                                     (ASLGEN'D)    82
 (R1)          (QANONHND) 001-1                                                                                     (ASLGEN'D)    83
 (R1)          (ORDQTY  ) 001-1                                                                                     (ASLGEN'D)    84
 (R1)          (UNITCOST) 001-1                                                                                     (ASLGEN'D)    85
 (R1)         (T,COSTVALU)                     (1)                                                                  (ASLGEN'D)    86
 (R1)          (ITMPRICE) 001-1                                                                                     (ASLGEN'D)    87
 (R1)         (T,SALEVALU)                     (1)                                                                  (ASLGEN'D)    88
------------------------------------------------------------------------------------------------------------------------------------
 STMT  SEQ                                                                                                           SEQUENCE   LIST
 TYPE  NO.                                        TEXT                                                               73 TO 80    SEQ
------------------------------------------------------------------------------------------------------------------------------------
 (T1)  (101)            (STOCK COST AND PRICE VALUE COMPARISON#                    )                                (ASLGEN'D)    89
1NOV 27, 2001  13.06.56                                                                                                  PAGE     14
                                                       *****************************
                                                       *  REQUEST NAME - REPT__13  *
                                                       *  GENERATED REQUEST        *
                                                       *****************************
------------------------------------------------------------------------------------------------------------------------------------
 STMT REPORT                          MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ         BACK BR RNT AB       SEQUENCE   LIST
 TYPE  DATE          REQUESTOR ID    ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM       73 TO 80    SEQ
------------------------------------------------------------------------------------------------------------------------------------
 (ER) (DATE  )                                                                (S)                               (ASLGEN'D)    90
------------------------------------------------------------------------------------------------------------------------------------
  ST SUM V  8  PG  PG   I SP  MAXIMUM TB COLUMN  DT  PG  STRT LIN SL RH .......SUBFILE........ C E A S R S N      C  SEQUENCE   LIST
  TY RPT S LPI WID HGT  M FRM LPP PGS TO HEADING POS POS PAGE NUM UB PN ENT     NAME   BLK/ FMT H F G T S U D      O  73 TO 80    SEQ
  MP     P              G             LT TYP POS         NUM      ML TD REC            BUF     C C S L T M S      N
------------------------------------------------------------------------------------------------------------------------------------
 (E1)                                                                                  (Y)                      (ASLGEN'D)    91
------------------------------------------------------------------------------------------------------------------------------------
           SP  Q                E  N  S  D  C  S  T  C  C  M  M  A  G  D Q PCT/RAT     P                 PARTIAL  SEQUENCE   LIST
 STMT SEQ  BF  L  FIELD   SEG-  N  P  E  E  T  U  O  U  N  A  I  V  M  E L  FIELD   RA- C     PICTURE    STRT NO  73 TO 80    SEQ
 TYPE NO.  COL F  NAME     LVL  D  R  Q  S  L  B  T  M  T  X  N  G  R  C F  NAME    TIO T      EDIT      CHR CHR
------------------------------------------------------------------------------------------------------------------------------------
 (R1)          (VENDOR  ) 001-1 (Y)   (1)   (1) (P)                                                                 (ASLGEN'D)    92
 (R1)         (T,PRODCODE)                  (1) (P)                                                                 (ASLGEN'D)    93
 (R1)          (ITEMNO  ) 001-1                                                                                     (ASLGEN'D)    94
 (R1)          (ITEMNAME) 001-1                                                                                     (ASLGEN'D)    95
 (R1)          (QANONHND) 001-1                                                                                     (ASLGEN'D)    96
 (R1)          (ORDQTY  ) 001-1                                                                                     (ASLGEN'D)    97
 (R1)          (UNITCOST) 001-1                                                                                     (ASLGEN'D)    98
 (R1)         (T,COSTVALU)                     (1)                                                                  (ASLGEN'D)    99
 (R1)          (ITMPRICE) 001-1                                                                                     (ASLGEN'D)   100
 (R1)         (T,SALEVALU)                     (1)                                                                  (ASLGEN'D)   101
------------------------------------------------------------------------------------------------------------------------------------
 STMT  SEQ                                                                                                           SEQUENCE   LIST
 TYPE  NO.                                        TEXT                                                               73 TO 80    SEQ
------------------------------------------------------------------------------------------------------------------------------------
 (T1)  (101)            (STOCK COST AND PRICE VALUE COMPARISON#                    )                                (ASLGEN'D)   102
1NOV 27, 2001  13.06.56                                                                                                  PAGE     15
                                                       *****************************
                                                       *  REQUEST NAME - REPT__14  *
                                                       *  GENERATED REQUEST        *
                                                       *****************************
------------------------------------------------------------------------------------------------------------------------------------
 STMT REPORT                          MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ         BACK BR RNT AB       SEQUENCE   LIST
 TYPE  DATE          REQUESTOR ID    ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM       73 TO 80    SEQ
------------------------------------------------------------------------------------------------------------------------------------
 (ER) (DATE  )                                                                (S)                               (ASLGEN'D)   103
------------------------------------------------------------------------------------------------------------------------------------
  ST SUM V  8  PG  PG   I SP  MAXIMUM TB COLUMN  DT  PG  STRT LIN SL RH .......SUBFILE........ C E A S R S N      C  SEQUENCE   LIST
  TY RPT S LPI WID HGT  M FRM LPP PGS TO HEADING POS POS PAGE NUM UB PN ENT     NAME   BLK/ FMT H F G T S U D      O  73 TO 80    SEQ
  MP     P              G             LT TYP POS         NUM      ML TD REC            BUF     C C S L T M S      N
------------------------------------------------------------------------------------------------------------------------------------
 (E1) (S)                                                                                                       (ASLGEN'D)   104
------------------------------------------------------------------------------------------------------------------------------------
           SP  Q                E  N  S  D  C  S  T  C  C  M  M  A  G  D Q PCT/RAT     P                 PARTIAL  SEQUENCE   LIST
 STMT SEQ  BF  L  FIELD   SEG-  N  P  E  E  T  U  O  U  N  A  I  V  M  E L  FIELD   RA- C     PICTURE    STRT NO  73 TO 80    SEQ
 TYPE NO.  COL F  NAME     LVL  D  R  Q  S  L  B  T  M  T  X  N  G  R  C F  NAME    TIO T      EDIT      CHR CHR
------------------------------------------------------------------------------------------------------------------------------------
 (R1)          (VENDOR  ) 001-1      (1)   (1)                                                                      (ASLGEN'D)   105
 (R1)         (T,PRODCODE)                 (1)                                                                      (ASLGEN'D)   106
 (R1)         (T,COSTVALU)                     (1) (G)                                                              (ASLGEN'D)   107
 (R1)         (T,SALEVALU)                     (1) (G)                                                              (ASLGEN'D)   108
------------------------------------------------------------------------------------------------------------------------------------
 STMT  SEQ                                                                                                           SEQUENCE   LIST
 TYPE  NO.                                        TEXT                                                               73 TO 80    SEQ
------------------------------------------------------------------------------------------------------------------------------------
 (T1)  (101)            (SUMMARY OF COST AND PRICE VALUE COMPARISON#               )                                (ASLGEN'D)   109
1NOV 27, 2001  13.06.56                                                                                                  PAGE     16
                                                       *****************************
                                                       *  REQUEST NAME - REPT__15  *
                                                       *  GENERATED REQUEST        *
                                                       *****************************
```

Figure D-39      Inventory Check List Report  (Page 6 of 13)

```
-------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                                MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID          ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM       73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(ER) (DATE  )                                                                    (S)                                (ASLGEN'D)   110
-------------------------------------------------------------------------------------------------------------------------------
 ST SUM V  8  PG  PG   I SP  MAXIMUM TB COLUMN  DT  PG  STRT LIN SL RH .......SUBFILE..... C E A S R S N      C  SEQUENCE   LIST
 TY RPT S LPI WID HGT  M FRM LPP PGS TO HEADING POS POS PAGE NUM UB PN ENT    NAME   BLK/ FMT H F G T S U D      O  73 TO 80    SEQ
 MP     P             G            LT TYP POS         NUM      ML TD REC          BUF      C C S L T M S      N
-------------------------------------------------------------------------------------------------------------------------------
(E1) (S)                                                        (X)                                            (ASLGEN'D)   111
-------------------------------------------------------------------------------------------------------------------------------
         SP  Q              E N S D C S T C C M M A G D Q PCT/RAT     P              PARTIAL   SEQUENCE   LIST
STMT SEQ BF  L  FIELD  SEG- N P E E T U O U N A I V M E L  FIELD  RA- C     PICTURE    STRT NO  73 TO 80    SEQ
TYPE NO. COL F  NAME     LVL D R Q S L B T M T X N G R C F  NAME   TIO T      EDIT      CHR CHR
-------------------------------------------------------------------------------------------------------------------------------
(R1)          (VENDOR  ) 001-1      (1)   (1)                                                                (ASLGEN'D)   112
(R1)        (T,PRODCODE)                  (1)                                                                (ASLGEN'D)   113
(R1)        (T,COSTVALU)                        (1) (G)                                                      (ASLGEN'D)   114
(R1)        (T,SALEVALU)                        (1) (G)                                                      (ASLGEN'D)   115
-------------------------------------------------------------------------------------------------------------------------------
STMT  SEQ                                                                                                    SEQUENCE   LIST
TYPE  NO.                                        TEXT                                                        73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
 (T1) (101)                (SUMMARY OF COST AND PRICE VALUE COMPARISON#              )                        (ASLGEN'D)   116
1NOV 27, 2001  13.06.56                                                                                        PAGE     17
                                          ******************************
                                          *  PROC NAME - OUTEDIT       *
                                          *  INPUT STREAM PROCEDURE    *
                                          ******************************
-------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                                MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID          ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM       73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
 (ER)                                                                            (N)                                (ASLGEN'D)   117
0     OUTEDIT: PROC
      LET T.NEWCOST  = 1.09 * UNITCOST
      LET T.NEWPRICE = 1.15 * ITMPRICE
      LET T.DIFCOST  = T.NEWCOST - UNITCOST
      LET T.DIFPRICE = T.NEWPRICE - ITMPRICE
      ;
       REPORT  PF(ITEMNO,1,2), VENDOR, ITEMNO, ITEMNAME,
               UNITCOST, T.NEWCOST, ITMPRICE, T.NEWPRICE
          ITEM PF(ITEMNO,1,2) PICTURE P'PRODUCT GROUP ##'
          ITEM VENDOR         PICTURE P'VENDOR ###############'
          FORMAT DATEEMT DATE
          TITLE 'RECOMMENDED NEW DOCUMENT PRICES'
          TITLE 'PRODUCT BY VENDOR AND ITEM'
          ORDER BY PF(ITEMNO,1,2) VENDOR ITEMNO ITEMNAME
          GROUP BY PF(ITEMNO,1,2) SUBTITLE NEWPAGE
          GROUP BY VENDOR         SUBTITLE
       END REPORT
      ;f272
       REPORT  PF(ITEMNO,1,2), ITEMNO, ITEMNAME,
               T.NEWCOST, T.NEWPRICE, T.DIFCOST, T.DIFPRICE
          ITEM PF(ITEMNO,1,2) PICTURE P'PRODUCT GROUP ##'
          ITEM T.DIFCOST  NONPRINT PICTURE P'$$$,$$$,$$9.99' SPACES 5
          ITEM T.DIFPRICE NONPRINT PICTURE P'$$$,$$$,$$9.99' SPACES 5
          FORMAT DATEFMT DATE
          TITLE 'ESTIMATED COST AND PRICE'
          TITLE 'VALUES AND DIFFERENCES'
          ORDER BY PF(ITEMNO,1,2)
          GROUP BY PF(ITEMNO,1,2) SUBTITLE NEWPAGE
          CUMULATE T.DIFCOST  AT G
          CUMULATE T.DIFPRICE AT G
          TOTAL T.DIFCOST  BY PF(ITEMNO,1,2)
          TOTAL T.DIFPRICE BY PF(ITEMNO,1,2)
       END REPORT
      ;f274
      END PROC
               ; editting output
-------------------------------------------------------------------------------------------------------------------------------
STMT SEQ  LOG CON .....OPERAND-A.....  OPER  .............OPERAND-B.............   ......RESULT......  PARTIAL FIELD  SEQUENCE   LIST
TYPE NO.  LEV CTR QLF FIELD   SEG-LVL  ATN   QLF FIELD   SEG-LVL  (OR CONSTANT)   QLF FIELD  SEG-LVL  STRT NUMB OPR  73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(PR) (100)                             (R )   (D,1.09                          ) (T,PIMP_22)                        (ASLGEN'D)   118
** MK4DI02  TYPE 0  DEFINITION OF TEMPORARY FIELD PIMP_22 - FORMAT P  LENGTH   8  DECIMALS 2 EDIT LENGTH  21 .                 *****
(PR) (101)          (T,PIMP_22)        (* )     (UNITCOST) 001-1                 (T,NEWCOST )                         (ASLGEN'D)   119
(PR) (102)                             (R )   (D,1.15                          ) (T,PIMP_24)                        (ASLGEN'D)   120
** MK4DI02  TYPE 0  DEFINITION OF TEMPORARY FIELD PIMP_24 - FORMAT P  LENGTH   8  DECIMALS 2 EDIT LENGTH  21 .                 *****
(PR) (103)          (T,PIMP_24)        (* )     (ITMPRICE) 001-1                 (T,NEWPRICE)                         (ASLGEN'D)   121
(PR) (104)          (T,NEWCOST )       (- )     (UNITCOST) 001-1                 (T,DIFCOST )                         (ASLGEN'D)   122
(PR) (105)          (T,NEWPRICE)       (- )     (ITMPRICE) 001-1                 (T,DIFPRICE)                         (ASLGEN'D)   123
1NOV 27, 2001  13.06.56                                                                                              PAGE     18
-------------------------------------------------------------------------------------------------------------------------------
STMT SEQ  LOG CON .....OPERAND-A.....  OPER  .............OPERAND-B.............   ......RESULT......  PARTIAL FIELD  SEQUENCE   LIST
```

Figure D-39 Inventory Check List Report (Page 7 of 13)

```
TYPE NO.  LEV CTR QLF FIELD   SEG-LVL  ATN   QLF FIELD   SEG-LVL  (OR CONSTANT)   QLF FIELD  SEG-LVL  STRT NUMB OPR  73 TO 80     SEQ
-------------------------------------------------------------------------------------------------------------------------------------
 (PR) (106)                           (GO)     (SUB REPT__28                  )                                  (ASLGEN'D)   124
 (PR) (107)                           (GO)     (SUB REPT__29                  )                                  (ASLGEN'D)   125
1NOV 27, 2001  13.06.56                                                                                                PAGE    19
                                                   *****************************
                                                   *  REQUEST NAME - REPT__28  *
                                                   *  GENERATED REQUEST        *
                                                   *****************************
-------------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                                 MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID           ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM       73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------------
 (ER) (DATE  )                                                                  (S)                              (ASLGEN'D)   126
-------------------------------------------------------------------------------------------------------------------------------------
  ST SUM V  8  PG  PG   I SP  MAXIMUM TB COLUMN  DT  PG  STRT LIN SL RH .......SUBFILE........ C E A S R S N      C  SEQUENCE   LIST
  TY RPT S LPI WID HGT  M FRM LPP PGS TO HEADING POS POS PAGE NUM UB PN ENT     NAME   BLK/ FMT H F G T S U D      O  73 TO 80    SEQ
  MP     P              G              LT TYP POS         NUM      ML TD REC           BUF      C C S L T M S      N
-------------------------------------------------------------------------------------------------------------------------------------
 (E1)                                                                                                            (ASLGEN'D)   127
-------------------------------------------------------------------------------------------------------------------------------------
          SP  Q                E  N  S  D  C  S  T  C  C  M  M  A  G  D Q PCT/RAT      P                 PARTIAL  SEQUENCE   LIST
STMT SEQ  BF  L  FIELD   SEG-  N  P  E  E  T  U  O  U  N  A  I  V  M  E L  FIELD   RA- C      PICTURE    STRT NO  73 TO 80    SEQ
TYPE NO.  COL F  NAME      LVL D  R  Q  S  L  B  T  M  T  X  N  G  R  C F  NAME    TIO T       EDIT      CHR CHR
-------------------------------------------------------------------------------------------------------------------------------------
 (R1)          (ITEMNO  ) 001-1      (1)    (1) (P)                                                ( 1) ( 2)  (ASLGEN'D)   128
** MK4EB09  TYPE 0  THE EXTENDED EDIT PATTERN FOR FIELD  ,ITEMNO   WILL BE USED - PRODUCT GROUP ##             .               *****
 (R1)          (VENDOR  ) 001-1      (2)    (2) (S)                                                           (ASLGEN'D)   129
** MK4EB09  TYPE 0  THE EXTENDED EDIT PATTERN FOR FIELD  ,VENDOR   WILL BE USED - VENDOR ##############        .               *****
 (R1)          (ITEMNO  ) 001-1      (3)                                                                      (ASLGEN'D)   130
 (R1)          (ITEMNAME) 001-1      (4)                                                                      (ASLGEN'D)   131
 (R1)          (UNITCOST) 001-1                                                                               (ASLGEN'D)   132
 (R1)        (T,NEWCOST )                                                                                     (ASLGEN'D)   133
 (R1)          (ITMPRICE) 001-1                                                                               (ASLGEN'D)   134
 (R1)        (T,NEWPRICE)                                                                                     (ASLGEN'D)   135
-------------------------------------------------------------------------------------------------------------------------------------
STMT  SEQ                                                                                                       SEQUENCE   LIST
TYPE  NO.                                                 TEXT                                                  73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------------
 (T1)  (101)                (RECOMMENDED NEW DOCUMENT PRICES#                          )                     (ASLGEN'D)   136
 (T1)  (102)                (PRODUCT BY VENDOR AND ITEM#                               )                     (ASLGEN'D)   137
1NOV 27, 2001  13.06.56                                                                                                PAGE    20
                                                   *****************************
                                                   *  REQUEST NAME - REPT__29  *
                                                   *  GENERATED REQUEST        *
                                                   *****************************
-------------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                                 MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID           ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM       73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------------
 (ER) (DATE  )                                                                  (S)                              (ASLGEN'D)   138
-------------------------------------------------------------------------------------------------------------------------------------
  ST SUM V  8  PG  PG   I SP  MAXIMUM TB COLUMN  DT  PG  STRT LIN SL RH .......SUBFILE........ C E A S R S N      C  SEQUENCE   LIST
  TY RPT S LPI WID HGT  M FRM LPP PGS TO HEADING POS POS PAGE NUM UB PN ENT     NAME   BLK/ FMT H F G T S U D      O  73 TO 80    SEQ
  MP     P              G              LT TYP POS         NUM      ML TD REC           BUF      C C S L T M S      N
-------------------------------------------------------------------------------------------------------------------------------------
 (E1)                                                                                                            (ASLGEN'D)   139
-------------------------------------------------------------------------------------------------------------------------------------
          SP  Q                E  N  S  D  C  S  T  C  C  M  M  A  G  D Q PCT/RAT      P                 PARTIAL  SEQUENCE   LIST
STMT SEQ  BF  L  FIELD   SEG-  N  P  E  E  T  U  O  U  N  A  I  V  M  E L  FIELD   RA- C      PICTURE    STRT NO  73 TO 80    SEQ
TYPE NO.  COL F  NAME      LVL D  R  Q  S  L  B  T  M  T  X  N  G  R  C F  NAME    TIO T       EDIT      CHR CHR
-------------------------------------------------------------------------------------------------------------------------------------
 (R1)          (ITEMNO  ) 001-1      (1)    (1) (P)                                                ( 1) ( 2)  (ASLGEN'D)   140
** MK4EB09  TYPE 0  THE EXTENDED EDIT PATTERN FOR FIELD  ,ITEMNO   WILL BE USED - PRODUCT GROUP ##             .               *****
 (R1)          (ITEMNO  ) 001-1                                                                               (ASLGEN'D)   141
 (R1)          (ITEMNAME) 001-1                                                                               (ASLGEN'D)   142
 (R1)        (T,NEWCOST )                                                                                     (ASLGEN'D)   143
 (R1)        (T,NEWPRICE)                                                                                     (ASLGEN'D)   144
 (R1)    ( 5) (T,DIFCOST )          (Y)            (1) (G)                                                    (ASLGEN'D)   145
** MK4EB09  TYPE 0  THE EXTENDED EDIT PATTERN FOR FIELD T,DIFCOST  WILL BE USED - $$$,$$$,$$9.99               .               *****
 (R1)    ( 5) (T,DIFPRICE)          (Y)            (1) (G)                                                    (ASLGEN'D)   146
** MK4EB09  TYPE 0  THE EXTENDED EDIT PATTERN FOR FIELD T,DIFPRICE WILL BE USED - $$$,$$$,$$9.99               .               *****
-------------------------------------------------------------------------------------------------------------------------------------
STMT  SEQ                                                                                                       SEQUENCE   LIST
TYPE  NO.                                                 TEXT                                                  73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------------
 (T1)  (101)                (ESTIMATED COST AND PRICE#                                 )                     (ASLGEN'D)   147
 (T1)  (102)                (VALUES AND DIFFERENCES#                                   )                     (ASLGEN'D)   148
1NOV 27, 2001  13.06.56                                                                                                PAGE    21
** MK4W204  TYPE 0  NUMBER OF MESSAGES PRINTED IS    9.                                                                    *****
                    M4OLD    -      21 RECORDS INPUT
                    M4REPO   -     405 RECORDS OUTPUT
                    M4INPUT  -     160 RECORDS INPUT
                    M4LIST   -     719 RECORDS OUTPUT
                       5 TRACKS ASSIGNED TO M4LIB    --    4 TRACKS NOT FULL. LIBRARY DIRECTORY BLOCKING FACTOR IS  272.
** MK4W209  TYPE 0  1011096 BYTES OF MAIN STORAGE UNUSED DURING DECODING PHASE                                             *****
```

Figure D-39      Inventory Check List Report  (Page 8 of 13)

```
                    1010160 BYTES OF MAIN STORAGE UNUSED DURING COMPILATION PHASE
                     622432 BYTES OF MAIN STORAGE UNUSED DURING PROCESSING PHASE
1** MK4ED02  TYPE 0  REQUEST-INVCHK1   REPORT-1  REQUESTOR ID-                                                                *****
106/14/02                                                INVENTORY CHECKLIST                                          PAGE 1
0-------------------------------------------------------------------------------------------------------------------------------
   ITEM          ITEM          QTY ON  REORDER  ORDER   UNIT      ITEM        VENDOR       ORDER   QTY BACK
  NUMBER         NAME           HAND    POINT    QTY    COST      PRICE                     NUMBER  ORDERED
 ------------------------------------------------------------------------------------------------------------------------------
   BL-B01   BLDR QUICK REF         249     100     200    $7.00     $7.95  ACCESS MASTERS      -       -
   BL-E68   BUILDER CONCEPTS        87     100     200    20.25     27.50  ACCESS MASTERS  45097      90
   BL-F43   BLDR MESSAGES           16     100     200    12.25     21.00  ACCESS MASTERS  10081      50
   BL-F43   BLDR MESSAGES           16     100     200    12.25     21.00  ACCESS MASTERS  31565      50
   BL-H11A  BLDR HANDBOOK ASL       56     100     200    23.25     31.00  ACCESS MASTERS  10987      25
   BL-H11A  BLDR HANDBOOK ASL       56     100     200    23.25     31.00  ACCESS MASTERS  12568      50
   BL-H11F  BLDR HANDBOOK FIXED     50      50     100    23.75     35.00  ACCESS MASTERS  65012      60
   BL-T01B  BLDR TRAINING BASIC     15       5       5      .00    775.00  IN HOUSE        90311       2
   BL-T02A  BLDR TRAINING ADV       15       5       5      .00  1,000.00  IN HOUSE            -       -
   GN-N34   DATA BASE CONCEPTS      14      50      50    33.25     39.95  TOOM CONSULTING     -       -
   IN-E13   INFORM USER GUIDE      175     100     200    23.25     28.95  QUERY PRO       11011      15
   IN-G61   INFORM TRAINING        85     100     200    15.15     23.50  QUERY PRO       28422      15
   IN-G61   INFORM TRAINING        85     100     200    15.15     23.50  QUERY PRO       36576      50
   IN-J17   INFORM CONCEPTS       197     100     200    15.15     23.50  QUERY PRO       39531       5
   IN-J17   INFORM CONCEPTS       197     100     200    15.15     23.50  QUERY PRO       86071      10
   IN-M41   INFORM MESSAGES        62      75     150    20.00     28.95  QUERY PRO       21116      10
   IN-M41   INFORM MESSAGES        62      75     150    20.00     28.95  QUERY PRO       31012      35
   IN-N19C  INFORM INSTALL CICS    11      75     150    11.30     18.95  QUERY PRO       53227      20
   IN-N19C  INFORM INSTALL CICS    11      75     150    11.30     18.95  QUERY PRO       89003      35
   IN-N19I  INFORM INSTALL IMS     38      25      50    11.30     18.95  QUERY PRO       53226       9
   TR-M52   TRANSACT CONCEPTS      28      25      25    14.50     19.95  ONLINE WIZARD       -       -
   TR-M55   TRANSACT INSTALL       3      25      25     9.00     15.95  ONLINE WIZARD   70652      12
   TR-M89   TRANSACT MESSAGES      19      25      50    37.75     47.50  ONLINE WIZARD   72568      13
   TR-M89   TRANSACT MESSAGES      19      25      50    37.75     47.50  ONLINE WIZARD   79031       5
   TR-N94C  TRANSACT CICS GUIDE    30      25      25    23.25     39.95  ONLINE WIZARD       -       -
   TR-N94I  TRANSACT IMS GUIDE     15      25      25    23.25     39.95  ONLINE WIZARD   21455       3
   TR-O13   TRANSACT QUICK REF     59      25      50     6.25      7.95  ONLINE WIZARD       -       -
   TR-O17   TRANSACT INST UPDT     25      25      25    16.40     23.95  ONLINE WIZARD       -       -
1** MK4FT06  TYPE 1       14 FIELDS WERE PRINTED AS INVALID, MISSING OR OVERFLOW.                                          *****
 ** MK4FT03  TYPE 0  END OF REPORT.                                                                                        *****
1** MK4ED02  TYPE 0  REQUEST-ORDERPT   REPORT-1  REQUESTOR ID-                                                            *****
1JUN 14, 2002                                            ORDERING REPORT                                              PAGE 1
0-------------------------------------------------------------------------------------------------------------------------------
   ITEM          ITEM                QTY ON  REORDER          ORDER
  NUMBER         NAME                 HAND    POINT            QTY
 ------------------------------------------------------------------------------------------------------------------------------
   BL-B01   BLDR QUICK REF              249     100             200
   BL-E68   BUILDER CONCEPTS             87     100             200
   BL-F43   BLDR MESSAGES                16     100             200
   BL-H11A  BLDR HANDBOOK ASL            56     100             200
   BL-H11F  BLDR HANDBOOK FIXED          50      50             100
   BL-T01B  BLDR TRAINING BASIC          15       5               5
   BL-T02A  BLDR TRAINING ADV            15       5               5
   GN-N34   DATA BASE CONCEPTS           14      50              50
   IN-E13   INFORM USER GUIDE           175     100             200
   IN-G61   INFORM TRAINING              85     100             200
   IN-J17   INFORM CONCEPTS             197     100             200
   IN-M41   INFORM MESSAGES              62      75             150
   IN-N19C  INFORM INSTALL CICS          11      75             150
   IN-N19I  INFORM INSTALL IMS           38      25              50
   TR-M52   TRANSACT CONCEPTS            28      25              25
   TR-M55   TRANSACT INSTALL              3      25              25
   TR-M89   TRANSACT MESSAGES            19      25              50
   TR-N94C  TRANSACT CICS GUIDE          30      25              25
   TR-N94I  TRANSACT IMS GUIDE           15      25              25
   TR-O13   TRANSACT QUICK REF           59      25              50
   TR-O17   TRANSACT INST UPDT           25      25              25
1** MK4FT03  TYPE 0  END OF REPORT.                                                                                        *****
1** MK4ED02  TYPE 0  REQUEST-INVCOUNT  REPORT-1  REQUESTOR ID-                                                            *****
1                INVENTORY COUNT FORM      PAGE 1                    INVENTORY COUNT FORM      PAGE 2
0   ITEM         ITEM          IN    REORDER  ORDER         ITEM         ITEM          IN    REORDER  ORDER
   NUMBER        NAME         STOCK   POINT    QTY         NUMBER        NAME         STOCK   POINT    QTY
0  BL-B01   BLDR QUICK REF   _____     100     200         IN-N19C  INFORM INSTALL   _____     75      150
0  BL-E68   BUILDER CONCEPT  _____     100     200         IN-N19I  INFORM INSTALL   _____     25       50
0  BL-F43   BLDR MESSAGES    _____     100     200         TR-M52   TRANSACT CONCEP  _____     25       25
0  BL-H11A  BLDR HANDBOOK A  _____     100     200         TR-M55   TRANSACT INSTAL  _____     25       25
0  BL-H11F  BLDR HANDBOOK F  _____      50     100         TR-M89   TRANSACT MESSAG  _____     25       50
0  BL-T01B  BLDR TRAINING B  _____       5       5         TR-N94C  TRANSACT CICS G  _____     25       25
0  BL-T02A  BLDR TRAINING A  _____       5       5         TR-N94I  TRANSACT IMS GU  _____     25       25
0  GN-N34   DATA BASE CONCE  _____      50      50         TR-O13   TRANSACT QUICK   _____     25       50
0  IN-E13   INFORM USER GUI  _____     100     200         TR-O17   TRANSACT INST U  _____     25       25
0  IN-G61   INFORM TRAINING  _____     100     200
0  IN-J17   INFORM CONCEPTS  _____     100     200
0  IN-M41   INFORM MESSAGES  _____      75     150
1** MK4FT03  TYPE 0  END OF REPORT.                                                                                        *****
1** MK4ED02  TYPE 0  REQUEST-REPT__6  REPORT-1  REQUESTOR ID-                                                            *****
1JUN 14, 2002                                    STOCK COST AND PRICE VALUE COMPARISON                                PAGE 1
0-------------------------------------------------------------------------------------------------------------------------------
```

Figure D-39 Inventory Check List Report (Page 9 of 13)

```
   ITEM          ITEM              VENDOR        QTY ON  ORDER   UNIT      VALUE      ITEM      VALUE AT
  NUMBER         NAME                             HAND    QTY    COST     AT COST     PRICE    SALE PRICE
 ---------------------------------------------------------------------------------------------------------------
   BL-B01   BLDR QUICK REF        ACCESS MASTERS    249    200    $7.00   $1,743.00    $7.95   $1,979.55
   BL-E68   BUILDER CONCEPTS                         87    200    20.25    1,761.75    27.50    2,392.50
   BL-F43   BLDR MESSAGES                            16    200    12.25      196.00    21.00      336.00
   BL-H11A  BLDR HANDBOOK ASL                        56    200    23.25    1,302.00    31.00    1,736.00
   BL-H11F  BLDR HANDBOOK FIXED                      50    100    23.75    1,187.50    35.00    1,750.00
0  BL-T01B  BLDR TRAINING BASIC   IN HOUSE          15      5     $.00        $.00   $775.00  $11,625.00
   BL-T02A  BLDR TRAINING ADV                        15      5      .00         .00 1,000.00   15,000.00
0  TR-M52   TRANSACT CONCEPTS     ONLINE WIZARD     28     25   $14.50     $406.00   $19.95     $558.60
   TR-M55   TRANSACT INSTALL                         3     25     9.00       27.00    15.95      47.85
   TR-M89   TRANSACT MESSAGES                       19     50    37.75      717.25    47.50     902.50
   TR-N94C  TRANSACT CICS GUIDE                     30     25    23.25      697.50    39.95   1,198.50
   TR-N94I  TRANSACT IMS GUIDE                      15     25    23.25      348.75    39.95     599.25
   TR-O13   TRANSACT QUICK REF                      59     50     6.25      368.75     7.95     469.05
   TR-O17   TRANSACT INST UPDT                      25     25    16.40      410.00    23.95     598.75
0  IN-E13   INFORM USER GUIDE     QUERY PRO        175    200   $23.25   $4,068.75   $28.95   $5,066.25
   IN-G61   INFORM TRAINING                         85    200    15.15    1,287.75    23.50   1,997.50
   IN-J17   INFORM CONCEPTS                        197    200    15.15    2,984.55    23.50   4,629.50
   IN-M41   INFORM MESSAGES                        62    150    20.00    1,240.00    28.95   1,794.90
   IN-N19C  INFORM INSTALL CICS                    11    150    11.30      124.30    18.95     208.45
   IN-N19I  INFORM INSTALL IMS                     38     50    11.30      429.40    18.95     720.10
0  GN-N34   DATA BASE CONCEPTS    TCOM CONSULTING   14     50   $33.25     $465.50   $39.95     $559.30
1** MK4FT03  TYPE 0  END OF REPORT.                                                                    *****
1** MK4ED02  TYPE 0  REQUEST-REPT___7  REPORT-1  REQUESTOR ID-                                         *****
1JUN 14, 2002                          BACKORDERS BY VENDOR AND ITEM                          PAGE 1
0---------------------------------------------------------------------------------------------------------------
                    VENDOR        ITEM          ITEM          ORDER   QTY BACK
                                  NUMBER        NAME          NUMBER  ORDERED
 ---------------------------------------------------------------------------------------------------------------
                 ACCESS MASTERS   BL-E68   BUILDER CONCEPTS     45097       90
0ITEMNO   TOTAL  ACCESS MASTERS   BL-E68                                    90
0                                 BL-F43   BLDR MESSAGES        10081       50
                                           BLDR MESSAGES        31565       50
0ITEMNO   TOTAL  ACCESS MASTERS   BL-F43                                   100
0                                 BL-H11A  BLDR HANDBOOK ASL    10987       25
                                           BLDR HANDBOOK ASL    12568       50
0ITEMNO   TOTAL  ACCESS MASTERS   BL-H11A                                   75
0                                 BL-H11F  BLDR HANDBOOK FIXED  65012       60
0ITEMNO   TOTAL  ACCESS MASTERS   BL-H11F                                   60
0VENDOR   TOTAL  ACCESS MASTERS                                            325
0                 IN HOUSE        BL-T01B  BLDR TRAINING BASIC  90311        2
0ITEMNO   TOTAL  IN HOUSE         BL-T01B                                    2
0VENDOR   TOTAL  IN HOUSE                                                  2
0                 ONLINE WIZARD   TR-M55   TRANSACT INSTALL     70652       12
0ITEMNO   TOTAL  ONLINE WIZARD    TR-M55                                    12
0                                 TR-M89   TRANSACT MESSAGES    72568       13
                                           TRANSACT MESSAGES    79031        5
0ITEMNO   TOTAL  ONLINE WIZARD    TR-M89                                    18
0                                 TR-N94I  TRANSACT IMS GUIDE   21455        3
0ITEMNO   TOTAL  ONLINE WIZARD    TR-N94I                                    3
0VENDOR   TOTAL  ONLINE WIZARD                                             33
0                 QUERY PRO       IN-E13   INFORM USER GUIDE    11011       15
0ITEMNO   TOTAL  QUERY PRO        IN-E13                                    15
0                                 IN-G61   INFORM TRAINING      28422       15
                                           INFORM TRAINING      36576       50
0ITEMNO   TOTAL  QUERY PRO        IN-G61                                    65
0                                 IN-J17   INFORM CONCEPTS      39531        5
                                           INFORM CONCEPTS      86071       10
0ITEMNO   TOTAL  QUERY PRO        IN-J17                                    15
1JUN 14, 2002                          BACKORDERS BY VENDOR AND ITEM                          PAGE 2
0---------------------------------------------------------------------------------------------------------------
                    VENDOR        ITEM          ITEM          ORDER   QTY BACK
                                  NUMBER        NAME          NUMBER  ORDERED
 ---------------------------------------------------------------------------------------------------------------
0                 QUERY PRO       IN-M41   INFORM MESSAGES      21116       10
                                           INFORM MESSAGES      31012       35
0ITEMNO   TOTAL  QUERY PRO        IN-M41                                    45
0                                 IN-N19C  INFORM INSTALL CICS  53227       20
                                           INFORM INSTALL CICS  89003       35
0ITEMNO   TOTAL  QUERY PRO        IN-N19C                                   55
0                                 IN-N19I  INFORM INSTALL IMS  53226        9
0ITEMNO   TOTAL  QUERY PRO        IN-N19I                                    9
0VENDOR   TOTAL  QUERY PRO                                                 204
1** MK4FT03  TYPE 0  END OF REPORT.                                                                    *****
1** MK4ED02  TYPE 0  REQUEST-REPT__12  REPORT-1  REQUESTOR ID-                                         *****
1JUN 14, 2002                          STOCK COST AND PRICE VALUE COMPARISON                  PAGE 1
0---------------------------------------------------------------------------------------------------------------
                    VENDOR       PRODCODE  ITEM          ITEM          QTY ON  ORDER   UNIT      VALUE      ITEM      VALUE AT
                                           NUMBER        NAME           HAND    QTY    COST     AT COST     PRICE    SALE PRICE
 ---------------------------------------------------------------------------------------------------------------
                 ACCESS MASTERS      BL    BL-B01   BLDR QUICK REF       249    200    $7.00   $1,743.00    $7.95   $1,979.55
                                     BL    BL-E68   BUILDER CONCEPTS      87    200    20.25    1,761.75    27.50    2,392.50
                                     BL    BL-F43   BLDR MESSAGES         16    200    12.25      196.00    21.00      336.00
                                     BL    BL-H11A  BLDR HANDBOOK ASL     56    200    23.25    1,302.00    31.00    1,736.00
```

Figure D-39 Inventory Check List Report (Page 10 of 13)

```
                                BL    BL-H11F  BLDR HANDBOOK FIXED      50   100    23.75   1,187.50    35.00   1,750.00
0VENDOR   TOTAL  ACCESS MASTERS                                                              $6,190.25            $8,194.05
0                IN HOUSE       BL    BL-T01B  BLDR TRAINING BASIC      15     5     $.00       $.00   $775.00  $11,625.00
                                BL    BL-T02A  BLDR TRAINING ADV        15     5      .00        .00 1,000.00   15,000.00
0VENDOR   TOTAL  IN HOUSE                                                                    $.00            $26,625.00
0                ONLINE WIZARD  TR    TR-M52   TRANSACT CONCEPTS       28    25   $14.50    $406.00    $19.95    $558.60
                                TR    TR-M55   TRANSACT INSTALL         3    25     9.00      27.00     15.95      47.85
                                TR    TR-M89   TRANSACT MESSAGES       19    50    37.75     717.25     47.50     902.50
                                TR    TR-N94C  TRANSACT CICS GUIDE     30    25    23.25     697.50     39.95   1,198.50
                                TR    TR-N94I  TRANSACT IMS GUIDE      15    25    23.25     348.75     39.95     599.25
                                TR    TR-O13   TRANSACT QUICK REF      59    50     6.25     368.75      7.95     469.05
                                TR    TR-O17   TRANSACT INST UPDT      25    25    16.40     410.00     23.95     598.75
0VENDOR   TOTAL  ONLINE WIZARD                                                              $2,975.25            $4,374.50
0                QUERY PRO      IN    IN-E13   INFORM USER GUIDE      175   200   $23.25  $4,068.75    $28.95   $5,066.25
                                IN    IN-G61   INFORM TRAINING         85   200    15.15   1,287.75     23.50   1,997.50
                                IN    IN-J17   INFORM CONCEPTS        197   200    15.15   2,984.55     23.50   4,629.50
                                IN    IN-M41   INFORM MESSAGES         62   150    20.00   1,240.00     28.95   1,794.90
                                IN    IN-N19C  INFORM INSTALL CICS     11   150    11.30     124.30     18.95     208.45
                                IN    IN-N19I  INFORM INSTALL IMS      38    50    11.30     429.40     18.95     720.10
0VENDOR   TOTAL  QUERY PRO                                                                $10,134.75           $14,416.70
0                TCOM CONSULTING GN    GN-N34   DATA BASE CONCEPTS      14    50   $33.25    $465.50    $39.95    $559.30
0VENDOR   TOTAL  TCOM CONSULTING                                                             $465.50              $559.30
1** MK4FT03  TYPE 0  END OF REPORT.                                                                                 *****
1** MK4ED02  TYPE 0  REQUEST-REPT__13  REPORT-1  REQUESTOR ID-                                                      *****
1JUN 14, 2002                                STOCK COST AND PRICE VALUE COMPARISON                                PAGE 1
0  ACCESS MASTERS
   BL
0-------------------------------------------------------------------------------------------------------------------------
                ITEM          ITEM          QTY ON  ORDER   UNIT      VALUE      ITEM      VALUE AT
                NUMBER        NAME           HAND    QTY    COST     AT COST    PRICE     SALE PRICE
 ------------------------------------------------------------------------------------------------------------------------
                BL-B01   BLDR QUICK REF       249    200   $7.00   $1,743.00    $7.95   $1,979.55
                BL-E68   BUILDER CONCEPTS      87    200   20.25    1,761.75    27.50    2,392.50
                BL-F43   BLDR MESSAGES        16    200   12.25      196.00    21.00      336.00
                BL-H11A  BLDR HANDBOOK ASL     56    200   23.25    1,302.00    31.00    1,736.00
                BL-H11F  BLDR HANDBOOK FIXED   50    100   23.75    1,187.50    35.00    1,750.00
0VENDOR   TOTAL                                                    $6,190.25             $8,194.05
1JUN 14, 2002                                STOCK COST AND PRICE VALUE COMPARISON                                PAGE 2
0  IN HOUSE
   BL
0-------------------------------------------------------------------------------------------------------------------------
                ITEM          ITEM          QTY ON  ORDER   UNIT      VALUE      ITEM      VALUE AT
                NUMBER        NAME           HAND    QTY    COST     AT COST    PRICE     SALE PRICE
 ------------------------------------------------------------------------------------------------------------------------
                BL-T01B  BLDR TRAINING BASIC    15      5     $.00       $.00   $775.00  $11,625.00
                BL-T02A  BLDR TRAINING ADV      15      5      .00        .00 1,000.00   15,000.00
0VENDOR   TOTAL                                                       $.00            $26,625.00
1JUN 14, 2002                                STOCK COST AND PRICE VALUE COMPARISON                                PAGE 3
0  ONLINE WIZARD
   TR
0-------------------------------------------------------------------------------------------------------------------------
                ITEM          ITEM          QTY ON  ORDER   UNIT      VALUE      ITEM      VALUE AT
                NUMBER        NAME           HAND    QTY    COST     AT COST    PRICE     SALE PRICE
 ------------------------------------------------------------------------------------------------------------------------
                TR-M52   TRANSACT CONCEPTS     28     25   $14.50    $406.00    $19.95    $558.60
                TR-M55   TRANSACT INSTALL       3     25     9.00      27.00     15.95      47.85
                TR-M89   TRANSACT MESSAGES     19     50    37.75     717.25     47.50     902.50
                TR-N94C  TRANSACT CICS GUIDE   30     25    23.25     697.50     39.95   1,198.50
                TR-N94I  TRANSACT IMS GUIDE    15     25    23.25     348.75     39.95     599.25
                TR-O13   TRANSACT QUICK REF    59     50     6.25     368.75      7.95     469.05
                TR-O17   TRANSACT INST UPDT    25     25    16.40     410.00     23.95     598.75
0VENDOR   TOTAL                                                    $2,975.25             $4,374.50
1JUN 14, 2002                                STOCK COST AND PRICE VALUE COMPARISON                                PAGE 4
0  QUERY PRO
   IN
0-------------------------------------------------------------------------------------------------------------------------
                ITEM          ITEM          QTY ON  ORDER   UNIT      VALUE      ITEM      VALUE AT
                NUMBER        NAME           HAND    QTY    COST     AT COST    PRICE     SALE PRICE
 ------------------------------------------------------------------------------------------------------------------------
                IN-E13   INFORM USER GUIDE    175    200   $23.25  $4,068.75    $28.95   $5,066.25
                IN-G61   INFORM TRAINING       85    200   15.15    1,287.75    23.50    1,997.50
                IN-J17   INFORM CONCEPTS      197    200   15.15    2,984.55    23.50    4,629.50
                IN-M41   INFORM MESSAGES       62    150   20.00    1,240.00    28.95    1,794.90
                IN-N19C  INFORM INSTALL CICS   11    150   11.30      124.30    18.95      208.45
                IN-N19I  INFORM INSTALL IMS    38     50   11.30      429.40    18.95      720.10
0VENDOR   TOTAL                                                   $10,134.75            $14,416.70
1JUN 14, 2002                                STOCK COST AND PRICE VALUE COMPARISON                                PAGE 5
0  TCOM CONSULTING
   GN
0-------------------------------------------------------------------------------------------------------------------------
                ITEM          ITEM          QTY ON  ORDER   UNIT      VALUE      ITEM      VALUE AT
                NUMBER        NAME           HAND    QTY    COST     AT COST    PRICE     SALE PRICE
 ------------------------------------------------------------------------------------------------------------------------
                GN-N34   DATA BASE CONCEPTS    14     50   $33.25    $465.50    $39.95    $559.30
0VENDOR   TOTAL                                                      $465.50              $559.30
1JUN 14, 2002                                STOCK COST AND PRICE VALUE COMPARISON                                PAGE 6
```

Figure D-39 Inventory Check List Report (Page 11 of 13)

```
-
0-------------------------------------------------------------------------------------------------------------------------------
                ITEM          ITEM          QTY ON  ORDER   UNIT      VALUE      ITEM      VALUE AT
               NUMBER          NAME          HAND    QTY    COST     AT COST     PRICE    SALE PRICE
 -------------------------------------------------------------------------------------------------------------------------------
 GRAND    TOTAL                                                          $19,765.75            $54,169.55
1** MK4FT03  TYPE 0  END OF REPORT.                                                                                          *****
1** MK4ED02  TYPE 0  REQUEST-REPT__14  REPORT-1  REQUESTOR ID-                                                               *****
1JUN 14, 2002                              SUMMARY OF COST AND PRICE VALUE COMPARISON                                  PAGE 1
0-------------------------------------------------------------------------------------------------------------------------------
                    VENDOR       PRODCODE    VALUE      VALUE AT
                                            AT COST    SALE PRICE
 -------------------------------------------------------------------------------------------------------------------------------
 VENDOR   TOTAL  ACCESS MASTERS      BL      $6,190.25   $8,194.05
0VENDOR   TOTAL  IN HOUSE            BL          $.00   $26,625.00
0VENDOR   TOTAL  ONLINE WIZARD       TR      $2,975.25   $4,374.50
0VENDOR   TOTAL  QUERY PRO           IN     $10,134.75  $14,416.70
0VENDOR   TOTAL  TCOM CONSULTING     GN        $465.50     $559.30
0GRAND    CUM.                             $19,765.75  $54,169.55
1** MK4FT03  TYPE 0  END OF REPORT.                                                                                          *****
1** MK4ED02  TYPE 0  REQUEST-REPT__15  REPORT-1  REQUESTOR ID-                                                               *****
1JUN 14, 2002                              SUMMARY OF COST AND PRICE VALUE COMPARISON                                  PAGE 1
0-------------------------------------------------------------------------------------------------------------------------------
       VENDOR       PRODCODE    VALUE      VALUE AT
                               AT COST    SALE PRICE
 -------------------------------------------------------------------------------------------------------------------------------
   ACCESS MASTERS      BL      $6,190.25   $8,194.05
0  IN HOUSE            BL          $.00   $26,625.00
0  ONLINE WIZARD       TR      $2,975.25   $4,374.50
0  QUERY PRO           IN     $10,134.75  $14,416.70
0  TCOM CONSULTING     GN        $465.50     $559.30
0                           $19,765.75  $54,169.55
1** MK4FT03  TYPE 0  END OF REPORT.                                                                                          *****
1** MK4ED02  TYPE 0  REQUEST-REPT__28  REPORT-1  REQUESTOR ID-                                                               *****
1JUN 14, 2002                                RECOMMENDED NEW DOCUMENT PRICES                                         PAGE 1
                                                PRODUCT BY VENDOR AND ITEM

0  PRODUCT GROUP BL
0-------------------------------------------------------------------------------------------------------------------------------
    ITEM          ITEM          UNIT     EXPECTED     ITEM     RECOMMENDED
   NUMBER          NAME          COST     NEW COST     PRICE     NEW PRICE
 -------------------------------------------------------------------------------------------------------------------------------
0  VENDOR ACCESS MASTERS
0  BL-B01   BLDR QUICK REF        $7.00       $7.63      $7.95       $9.14
   BL-E68   BUILDER CONCEPTS      20.25       22.07      27.50       31.62
   BL-F43   BLDR MESSAGES         12.25       13.35      21.00       24.15
   BL-H11A  BLDR HANDBOOK ASL     23.25       25.34      31.00       35.65
   BL-H11F  BLDR HANDBOOK FIXED   23.75       25.88      35.00       40.25
0  VENDOR IN HOUSE
0  BL-T01B  BLDR TRAINING BASIC     $.00        $.00    $775.00     $891.25
   BL-T02A  BLDR TRAINING ADV        .00         .00   1,000.00    1,150.00
1JUN 14, 2002                                RECOMMENDED NEW DOCUMENT PRICES                                         PAGE 2
                                                PRODUCT BY VENDOR AND ITEM

0  PRODUCT GROUP GN
0-------------------------------------------------------------------------------------------------------------------------------
    ITEM          ITEM          UNIT     EXPECTED     ITEM     RECOMMENDED
   NUMBER          NAME          COST     NEW COST     PRICE     NEW PRICE
 -------------------------------------------------------------------------------------------------------------------------------
0  VENDOR TCOM CONSULTING
0  GN-N34   DATA BASE CONCEPTS    $33.25      $36.24     $39.95      $45.94
1JUN 14, 2002                                RECOMMENDED NEW DOCUMENT PRICES                                         PAGE 3
                                                PRODUCT BY VENDOR AND ITEM

0  PRODUCT GROUP IN
0-------------------------------------------------------------------------------------------------------------------------------
    ITEM          ITEM          UNIT     EXPECTED     ITEM     RECOMMENDED
   NUMBER          NAME          COST     NEW COST     PRICE     NEW PRICE
 -------------------------------------------------------------------------------------------------------------------------------
0  VENDOR QUERY PRO
0  IN-E13   INFORM USER GUIDE     $23.25      $25.34     $28.95      $33.29
   IN-G61   INFORM TRAINING       15.15       16.51      23.50       27.02
   IN-J17   INFORM CONCEPTS       15.15       16.51      23.50       27.02
   IN-M41   INFORM MESSAGES       20.00       21.80      28.95       33.29
   IN-N19C  INFORM INSTALL CICS   11.30       12.31      18.95       21.79
   IN-N19I  INFORM INSTALL IMS    11.30       12.31      18.95       21.79
1JUN 14, 2002                                RECOMMENDED NEW DOCUMENT PRICES                                         PAGE 4
                                                PRODUCT BY VENDOR AND ITEM

0  PRODUCT GROUP TR
0-------------------------------------------------------------------------------------------------------------------------------
    ITEM          ITEM          UNIT     EXPECTED     ITEM     RECOMMENDED
   NUMBER          NAME          COST     NEW COST     PRICE     NEW PRICE
 -------------------------------------------------------------------------------------------------------------------------------
0  VENDOR ONLINE WIZARD
0  TR-M52   TRANSACT CONCEPTS     $14.50      $15.80     $19.95      $22.94
   TR-M55   TRANSACT INSTALL        9.00        9.81      15.95       18.34
   TR-M89   TRANSACT MESSAGES      37.75       41.14      47.50       54.62
   TR-N94C  TRANSACT CICS GUIDE    23.25       25.34      39.95       45.94
   TR-N94I  TRANSACT IMS GUIDE     23.25       25.34      39.95       45.94
```

Figure D-39 Inventory Check List Report (Page 12 of 13)

```
   TR-O13   TRANSACT QUICK REF       6.25        6.81       7.95        9.14
   TR-O17   TRANSACT INST UPDT      16.40       17.87      23.95       27.54
1** MK4FT03  TYPE 0  END OF REPORT.                                                                                  *****
1** MK4ED02  TYPE 0  REQUEST-REPT__29  REPORT-1  REQUESTOR ID-                                                       *****
1JUN 14, 2002                                          ESTIMATED COST AND PRICE                                  PAGE 1
                                                        VALUES AND DIFFERENCES
0  PRODUCT GROUP BL
0-----------------------------------------------------------------------------------------------------------------------
                 ITEM            ITEM           EXPECTED   RECOMMENDED       ESTIMATED          ESTIMATED
                NUMBER           NAME           NEW COST    NEW PRICE        COST VALUE        PRICE VALUE
 ----------------------------------------------------------------------------------------------------------------------
                BL-B01   BLDR QUICK REF            $7.63       $9.14
                BL-E68   BUILDER CONCEPTS          22.07       31.62
                BL-F43   BLDR MESSAGES             13.35       24.15
                BL-H11A  BLDR HANDBOOK ASL         25.34       35.65
                BL-H11F  BLDR HANDBOOK FIXED       25.88       40.25
                BL-T01B  BLDR TRAINING BASIC         .00      891.25
                BL-T02A  BLDR TRAINING ADV           .00    1,150.00
0ITEMNO   TOTAL                                                                  $7.77            $284.61
1JUN 14, 2002                                          ESTIMATED COST AND PRICE                                  PAGE 2
                                                        VALUES AND DIFFERENCES
0  PRODUCT GROUP GN
0-----------------------------------------------------------------------------------------------------------------------
                 ITEM            ITEM           EXPECTED   RECOMMENDED       ESTIMATED          ESTIMATED
                NUMBER           NAME           NEW COST    NEW PRICE        COST VALUE        PRICE VALUE
 ----------------------------------------------------------------------------------------------------------------------
                GN-N34   DATA BASE CONCEPTS       $36.24      $45.94
0ITEMNO   TOTAL                                                                  $2.99             $5.99
1JUN 14, 2002                                          ESTIMATED COST AND PRICE                                  PAGE 3
                                                        VALUES AND DIFFERENCES
0  PRODUCT GROUP IN
0-----------------------------------------------------------------------------------------------------------------------
                 ITEM            ITEM           EXPECTED   RECOMMENDED       ESTIMATED          ESTIMATED
                NUMBER           NAME           NEW COST    NEW PRICE        COST VALUE        PRICE VALUE
 ----------------------------------------------------------------------------------------------------------------------
                IN-E13   INFORM USER GUIDE        $25.34      $33.29
                IN-G61   INFORM TRAINING           16.51       27.02
                IN-J17   INFORM CONCEPTS           16.51       27.02
                IN-M41   INFORM MESSAGES           21.80       33.29
                IN-N19C  INFORM INSTALL CICS       12.31       21.79
                IN-N19I  INFORM INSTALL IMS        12.31       21.79
0ITEMNO   TOTAL                                                                  $8.63            $21.40
1JUN 14, 2002                                          ESTIMATED COST AND PRICE                                  PAGE 4
                                                        VALUES AND DIFFERENCES
0  PRODUCT GROUP TR
0-----------------------------------------------------------------------------------------------------------------------
                 ITEM            ITEM           EXPECTED   RECOMMENDED       ESTIMATED          ESTIMATED
                NUMBER           NAME           NEW COST    NEW PRICE        COST VALUE        PRICE VALUE
 ----------------------------------------------------------------------------------------------------------------------
                TR-M52   TRANSACT CONCEPTS        $15.80      $22.94
                TR-M55   TRANSACT INSTALL           9.81       18.34
                TR-M89   TRANSACT MESSAGES         41.14       54.62
                TR-N94C  TRANSACT CICS GUIDE       25.34       45.94
                TR-N94I  TRANSACT IMS GUIDE        25.34       45.94
                TR-O13   TRANSACT QUICK REF         6.81        9.14
                TR-O17   TRANSACT INST UPDT        17.87       27.54
0ITEMNO   TOTAL                                                                 $11.71            $29.26
 GRAND    CUM.                                                                  $31.10           $341.26
1** MK4FT03  TYPE 0  END OF REPORT.                                                                                  *****
1** MK4W204  TYPE 0  NUMBER OF MESSAGES PRINTED IS   23.                                                             *****
                    M4REPO   -      404 RECORDS INPUT
                    M4LIST   -      403 RECORDS OUTPUT
 ** MK4W209  TYPE 0  1055664 BYTES OF MAIN STORAGE UNUSED DURING REPORTING PHASE                                     *****
```

Figure D-39 Inventory Check List Report (Page 13 of 13)

# Arrays

## Preparation Job

```
IDCAMS  SYSTEM SERVICES                                    TIME: 16:56:38       11/05/01     PAGE      1

      DELETE (BUILDER.R140.SALESAMT.R13#1) CLUSTER    /* <<< HLQ */     01120001
IDC0550I ENTRY (D) BUILDER.R140.SALESAMT.R13#1.DATA DELETED
IDC0550I ENTRY (I) BUILDER.R140.SALESAMT.R13#1.INDEX DELETED
IDC0550I ENTRY (C) BUILDER.R140.SALESAMT.R13#1 DELETED
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0

      DELETE (BUILDER.R140.ARR2001.R13#1)  CLUSTER    /* <<< HLQ */     01120001
IDC0550I ENTRY (D) BUILDER.R140.ARR2001.R13#1.DATA DELETED
IDC0550I ENTRY (I) BUILDER.R140.ARR2001.R13#1.INDEX DELETED
IDC0550I ENTRY (C) BUILDER.R140.ARR2001.R13#1 DELETED
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0

      DELETE (BUILDER.R140.ARR2002.R13#1)  CLUSTER    /* <<< HLQ */     01120001
IDC0550I ENTRY (D) BUILDER.R140.ARR2002.R13#1.DATA DELETED
IDC0550I ENTRY (I) BUILDER.R140.ARR2002.R13#1.INDEX DELETED
IDC0550I ENTRY (C) BUILDER.R140.ARR2002.R13#1 DELETED
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0

      SET MAXCC=0

      DEFINE CLUSTER -                                             01140000
               (NAME('BUILDER.R140.SALESAMT.R13#1') -              01150001
               INDEXED KEYS (7,0) -
               VOL(ISPDA) -                                        01160000
               TRK(3 1) -
               SHAREOPTIONS(2 3)  -                                01180000
               RECORDSIZE(50 50)) -                                01220000
               DATA (CISZ(4096))          /* <<< NAME ... VOL       */
IGD01008I ALLOCATION SET TO IMDCLASS 897
IGD01010I ALLOCATION SET TO IMDVOLS 873
IDC0508I DATA ALLOCATION STATUS FOR VOLUME ISP806 IS 0
IDCAMS  SYSTEM SERVICES                                    TIME: 16:56:38       11/05/01     PAGE      2
IDC0509I INDEX ALLOCATION STATUS FOR VOLUME ISP806 IS 0
IDC0512I NAME GENERATED-(D) BUILDER.R140.SALESAMT.R13#1.DATA
IDC0512I NAME GENERATED-(I) BUILDER.R140.SALESAMT.R13#1.INDEX
IDC0181I STORAGECLASS USED IS IMDCLASS
IDC0181I MANAGEMENTCLASS USED IS DEFAULT
IDC0181I DATACLASS USED IS DEFAULT
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0

      DEFINE CLUSTER -                                             01140000
               (NAME('BUILDER.R140.ARR2001.R13#1') -               01150001
               INDEXED KEYS (4,0) -
               VOL(ISPDA) -                                        01160000
               TRK(3 1) -                                          01170001
               SHAREOPTIONS(2 3)  -                                01180000
               RECORDSIZE(7492 7492)) -                            01220000
               DATA (CISZ(8192))          /* <<< NAME ... VOL       */
IGD01008I ALLOCATION SET TO IMDCLASS 897
IGD01010I ALLOCATION SET TO IMDVOLS 873
IDC0508I DATA ALLOCATION STATUS FOR VOLUME ISP807 IS 0
IDC0509I INDEX ALLOCATION STATUS FOR VOLUME ISP807 IS 0
IDC0512I NAME GENERATED-(D) BUILDER.R140.ARR2001.R13#1.DATA
IDC0512I NAME GENERATED-(I) BUILDER.R140.ARR2001.R13#1.INDEX
IDC0181I STORAGECLASS USED IS IMDCLASS
IDC0181I MANAGEMENTCLASS USED IS DEFAULT
IDC0181I DATACLASS USED IS DEFAULT
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0

      DEFINE CLUSTER -                                             01140000
               (NAME('BUILDER.R140.ARR2002.R13#1') -               01150001
               INDEXED KEYS (4,0) -
               VOL(ISPDA) -                                        01160000
               TRK(3 1) -                                          01170001
               SHAREOPTIONS(2 3)  -                                01180000
               RECORDSIZE(7492 7492)) -                            01220000
               DATA (CISZ(8192))          /* <<< NAME ... VOL       */
IGD01008I ALLOCATION SET TO IMDCLASS 897
IGD01010I ALLOCATION SET TO IMDVOLS 873
IDCAMS  SYSTEM SERVICES                                    TIME: 16:56:38       11/05/01     PAGE      3
IDC0508I DATA ALLOCATION STATUS FOR VOLUME ISP804 IS 0
IDC0509I INDEX ALLOCATION STATUS FOR VOLUME ISP804 IS 0
IDC0512I NAME GENERATED-(D) BUILDER.R140.ARR2002.R13#1.DATA
IDC0512I NAME GENERATED-(I) BUILDER.R140.ARR2002.R13#1.INDEX
IDC0181I STORAGECLASS USED IS IMDCLASS
```

Figure D-40 Preparation Job (Page 1 of 9)

```
IDC0181I MANAGEMENTCLASS USED IS DEFAULT
IDC0181I DATACLASS USED IS DEFAULT
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0

IDC0002I IDCAMS PROCESSING COMPLETE. MAXIMUM CONDITION CODE WAS 0

NOV 05, 2001  16.56.48                                                                                    PAGE      1
                                        ******************************************
                                        *       COMLIB LIBRARY UTILITIES         *
                                        *       COPYRIGHT (C) 1995, 1997         *
                                        * COMPUTER ASSOCIATES INTERNATIONAL, INC. *
                                        ******************************************

                          DIRECTORY BLOCKING FACTOR:   0   LIBRARY RESERVE: NO   LIBRARY COMPONENT: C4.5
                                    COMPRESSION ACTIVE: NO    COMPRESSION SIZE MINIMUM:   507
** CLSUC01  TYPE 0  THE LIBRARY HAS BEEN INITIALIZED.                                                                *****

NOV 05, 2001  16.56.48                                                                                    PAGE      2
** MK4W204  TYPE 0  NUMBER OF MESSAGES PRINTED IS    1.                                                             *****
                  M4LIST   -          12 RECORDS OUTPUT
                    15 TRACKS ASSIGNED TO M4LIB    --   13 TRACKS NOT FULL. LIBRARY DIRECTORY BLOCKING FACTOR IS  800.
** MK4W209  TYPE 0   825880 BYTES OF MAIN STORAGE UNUSED DURING DICTIONARY MAINTENANCE                               *****

NOV 05, 2001  16.56.48                                                                                    PAGE      1
                                        ******************************************
                                        * VISION:Builder   4400  (OS/390 - 14.0)  *
                                        *          COPYRIGHT (C) 2002            *
                                        * COMPUTER ASSOCIATES INTERNATIONAL, INC. *
                                        ******************************************
VISION:Builder JCR1 14.0 110201              ENABLED FOR IBM LANGUAGE ENVIRONMENT                BUILD STAMP = 101306,12:51:24.
                                          COMLIB  MODIFICATIONS (PTFs, APARs) INSTALLED
                                       201,202,203,204,205,206,207,208,209,210,212,213,214,215,216
=============================================================================================================================
=                                                                                                                           =
=                  I N S T A L L A T I O N    P A R A M E T E R S    ( M 4 P A R A M S , M A R K L I B P )                 =
=                                                                                                                           =
=  SYSTEM DELIMITER:   #          PAGE HEIGHT:        66         M4LIST WIDTH:       132        DEF WIDTH OF PAGE:  0     =
=                                                                                                                           =
=  AUTO GRAND:         N          HEADING CHAR:       -          SUBTITLE REPEAT:    N          INVALID FIELD:      *     =
=                                                                                                                           =
=  MISSING FIELD:      -          NON-EDIT FIELD:     +          PERCENT CHAR:       %          LEFT SEPARATOR:     (     =
=                                                                                                                           =
=  RIGHT SEPARATOR:    )          SINGLE SEPARATOR:   ,          SOURCE SPACING:     1          PRINT MESSAGES:     Y     =
=                                                                                                                           =
=  CONSOLE MESSAGES:   N          M4REPO BLOCKSIZE:   4,096      INPUT I/O BUFFERS:  2          OUTPUT I/O BUFFERS: 1     =
=                                                                                                                           =
=  SNGL-STEP STORAGE:  8,192      SNGL-STEP SORTSIZE: 524,288    DIGIT SELECT CHAR:  9          ZERO SUPPRESS CHAR: Z     =
=                                                                                                                           =
=  CURRENCY CHAR:      $          PLUS CHAR:          +          MINUS CHAR:         -          CHECK PROTECT CHAR: *     =
=                                                                                                                           =
=  DECIMAL CHAR:       .          GROUPING CHAR:      ,          PRIMARY PLOT CHAR:  X          SECONDARY PLOT CHAR:*     =
=                                                                                                                           =
=  FIT PLOT CHAR:      .          HORIZONTAL AXIS:    _          HORIZONTAL HASH:    |          VERTICAL AXIS:      |     =
=                                                                                                                           =
=  VERTICAL HASH:      -          MINUTES/HOUR:       60         SECONDS/MINUTE:     60         TIME DELIMITER:     HH:MM:SS  =
=                                                                                                                           =
=  DATE FORMAT:      MM DD, YYYY  TODAY FORMAT+DELIM: MM/DD/YY   ISDATE DELIMITER:   YYYY-MM-DD  JULIAN DELIMITER:   YY.DDD   =
=                                                                                                                           =
=  SORT PROGRAM CODE:  2          MINCORE VALUE:      12 K       ALT M4LIST WIDTH:   264        ALT DEF W/OF PAGE:  0     =
=                                                                                                                           =
=  MAX LINES OF TRACE: 1,024      ITEM TRACKING:      0          SUPPRESS NDS REPT?: N          DEFAULT MAXGETMN:   1,024 K   =
=                                                                                                                           =
=  CONDITION CODE 1:   0          CONDITION CODE 2:   4          CONDITION CODE 3:   0          CONDITION CODE 4:   16    =
=                                                                                                                           =
=============================================================================================================================
NOV 05, 2001  16.56.48                                                                                    PAGE      2
-----------------------------------------------------------------------------------------------------------------------------
   RUN     STMT    FILE    O N S D U T R A B S D S L  RB  R S M R S          START          END           SEQUENCE   LIST
   NAME    TYPE    NAME    L E E I P R E U U R L C S  PL  J O O F S          SEARCH         SEARCH        73 TO 80   SEQ
                           D W Q R D N P D F T M N T  TK  T P P O R
-----------------------------------------------------------------------------------------------------------------------------
(DEFINE  ) (RC)                                                                                                          1
NOV 05, 2001  16.56.48                                                                                    PAGE      3
** CLSAA00  TYPE 0  DEFINITION WORKHRS  IS BEING PROCESSED.                                                        *****
-----------------------------------------------------------------------------------------------------------------------------
   FILE    STMT    FILE    DELETE GLOSSARY RECORD RECORD RECORDS BUFFER MAPPED     MAPPING        MAPPED        SEQUENCE   LIST
   NAME    TYPE    IDENT    FILE   LEVEL   FORMAT  SIZE  PER BLK  SIZE   FILE    INIT     TERM  RECSIZ        73 TO 80   SEQ
-----------------------------------------------------------------------------------------------------------------------------
(WORKHRS ) (FD)             (R)     (B)     (F)    (  80)  (100)  ( 8000)                                              2

NOV 05, 2001  16.56.48                                                                                    PAGE      4
                                      DETAILED GLOSSARY BY LOCATION FOR                                GLOSSARY PAGE     1
                                         FILE DEFINITION - WORKHRS
DATE CATALOGED - 11/05/01 16:56:48                          DATE LAST UPDATED - 11/05/01 16:56:48
CATALOGED BY -                                              LAST UPDATED BY -
EXPIRATION DATE -                                           DATE LAST USED -
```

Figure D-40 Preparation Job (Page 2 of 9)

```
FILE IDENTIFICATION =                                                    RECORD FORMAT = FIXED BLOCKED
NUMBER OF SEGMENTS IN FILE = 1                                           RECORD SIZE = 80
NUMBER OF FIELDS IN FILE = 5                                             BLOCK SIZE = 8000

                                    ********************************
                                    *    SEGMENT   1, LEVEL 1      *
                                    *   SEGMENT NAME = HOURS       *
                                    ********************************

SEGMENT OCCURS N TIMES = 1                                KEY FIELD 1 = STOREID    TYPE = C   LENGTH = 5
SEGMENT SIZE = 17
NUMBER OF FIELDS IN SEGMENT = 5
-------------------------------------------------------------------------------------------------------------------------------
FIELD   FIELD   FIELD    FIELD   FIELD    DEC    CNT FIELD  EDIT CODES     EDIT    OUTPUT   LINE  DATA BASE NAME/
NAME    TYPE   LOCATION  LENGTH  RNDING  PLACES  FOR SGMT   ( ) ( ) ( )   LENGTH   WIDTH    NO   *** COLUMN HEADING ***
-------------------------------------------------------------------------------------------------------------------------------
STOREID  C        1        5                                                 5        6
                                                                                      1    ***     STORE      ***
                                                                                      2    ***     ID NO.     ***
YEAR     C        6        4                                                 4        4
                                                                                      1    ***      YEAR      ***
WEEKNO   C       10        2                                                 2        4
                                                                                      1    ***      WEEK      ***
                                                                                      2    ***      NO.       ***
DAYNO    C       12        1                                                 1        3
                                                                                      1    ***      DAY       ***
                                                                                      2    ***      NO.       ***
HRSWRKD  Z       13        5          1                                      8        8
                                                                                      1    ***     HOURS      ***
                                                                                      2    ***     WORKED     ***

** CLSGB00  TYPE 0  END OF FILE DEFINITION GLOSSARY.                                                              *****
NOV 05, 2001  16.56.48                                                                                  PAGE     5
** CLSAA00  TYPE 0  DEFINITION SALEAMTS IS BEING PROCESSED.                                                       *****
-------------------------------------------------------------------------------------------------------------------------------
   FILE    STMT    FILE    DELETE GLOSSARY RECORD RECORD RECORDS BUFFER MAPPED      MAPPING         MAPPED          SEQUENCE   LIST
   NAME    TYPE    IDENT    FILE   LEVEL   FORMAT  SIZE  PER BLK  SIZE  FILE     INIT     TERM     RECSIZ         73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(SALEAMTS) (FD)             (R)     (B)     (K)                (   50)                                                  18

NOV 05, 2001  16.56.48                                                                                  PAGE     6
                                        DETAILED GLOSSARY BY LOCATION FOR                       GLOSSARY PAGE     1
                                           FILE DEFINITION - SALEAMTS
DATE CATALOGED - 11/05/01 16:56:48                                       DATE LAST UPDATED - 11/05/01 16:56:48
CATALOGED BY -                                                           LAST UPDATED BY -
EXPIRATION DATE -                                                        DATE LAST USED -
FILE IDENTIFICATION =                                                    RECORD FORMAT = KEY SEQUENCED VSAM
NUMBER OF SEGMENTS IN FILE = 1                                           RECORD SIZE = 50
NUMBER OF FIELDS IN FILE = 5                                             BUFFER SIZE = 50

                                    ********************************
                                    *    SEGMENT   1, LEVEL 1      *
                                    *   SEGMENT NAME = SALES       *
                                    ********************************

SEGMENT OCCURS N TIMES = 1                                KEY FIELD 1 = KEYFIELD   TYPE = C   LENGTH = 7
SEGMENT SIZE = 12
NUMBER OF FIELDS IN SEGMENT = 5
SEGMENT KEY CHARACTERISTICS = U
-------------------------------------------------------------------------------------------------------------------------------
FIELD   FIELD   FIELD    FIELD   FIELD    DEC    CNT FIELD  EDIT CODES     EDIT    OUTPUT   LINE  DATA BASE NAME/
NAME    TYPE   LOCATION  LENGTH  RNDING  PLACES  FOR SGMT   ( ) ( ) ( )   LENGTH   WIDTH    NO   *** COLUMN HEADING ***
-------------------------------------------------------------------------------------------------------------------------------
KEYFIELD C        1        7                                                 7        8
                                                                                      1    ***    WEEK/DAY    ***
                                                                                      2    ***      NO.       ***
YEAR     C        1        4                                                 4        4
                                                                                      1    ***      YEAR      ***
WEEKNO   C        5        2                                                 2        4
                                                                                      1    ***      WEEK      ***
                                                                                      2    ***      NO.       ***
DAYNO    C        7        1                                                 1        3
                                                                                      1    ***      DAY       ***
                                                                                      2    ***      NO.       ***
SALES    Z        8        5          1                                      8        8
                                                                                      1    ***     SALES      ***
                                                                                      2    ***     1000'S     ***

** CLSGB00  TYPE 0  END OF FILE DEFINITION GLOSSARY.                                                              *****
NOV 05, 2001  16.56.48                                                                                  PAGE     7
** CLSAA00  TYPE 0  DEFINITION COSTARR  IS BEING PROCESSED.                                                       *****
-------------------------------------------------------------------------------------------------------------------------------
   ARRAY   STMT         DELETE GLOSSARY RECORD RECORD RECORDS BUFFER   NUMBER   NUMBER                          SEQUENCE   LIST
   NAME    TYPE         ARRAY?  TYPE    FORMAT  SIZE  PER BLK  SIZE   OF ROWS  OF COLS                        73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(COSTARR ) (AD)          (R)     (B)     (K)                ( 7492) (    52) (     8)                                  34

NOV 05, 2001  16.56.48                                                                                  PAGE     8
                                        DETAILED GLOSSARY BY LOCATION FOR                       GLOSSARY PAGE     1
                                           ARRAY DEFINITION - COSTARR
```

Figure D-40 Preparation Job (Page 3 of 9)

```
DATE CATALOGED - 11/05/01 16:56:48                          DATE LAST UPDATED - 11/05/01 16:56:48
CATALOGED BY -                                              LAST UPDATED BY -
EXPIRATION DATE -                                           DATE LAST USED -
NUMBER OF ROWS = 52                                         RECORD FORMAT = KEY SEQUENCED VSAM
NUMBER OF COLUMNS = 8                                       RECORD SIZE = 7492          (COMPUTED)
DATA CELL SIZE = 18                                         BUFFER SIZE = 7492
ARRAY SIZE = 7492
-------------------------------------------------------------------------------------------------------------------------------
FIELD     FIELD    FIELD  FIELD  FIELD    DEC     ARRAY     EDIT CODES     EDIT    OUTPUT   LINE       REPORT FIELD
NAME      LOCATION LENGTH TYPE   RNDING   PLACES  KEY       ( ) ( ) ( )    LENGTH  WIDTH    NO  ***    HEADING    ***
-------------------------------------------------------------------------------------------------------------------------------
YEAR         1       4     C                        1                      4       4
                                                                                      1   ***    YEAR       ***
                                                                                      2   ***    NO.        ***
-------------------------------------------------------------------------------------------------------------------------------
WEEKNO       1       2     C                                               2       4
                                                                                      1   ***    WEEK       ***
                                                                                      2   ***    NO.        ***
DAYNO        3       1     C                                               1       3
                                                                                      1   ***    DAY        ***
                                                                                      2   ***    NO.        ***
HOURS        4       5     Z                1                              8       8
                                                                                      1   ***    HOURS      ***
                                                                                      2   ***    WORKED     ***
COSTS        9       5     Z                1                              8       8
                                                                                      1   ***    COSTS      ***
                                                                                      2   ***    1000'S     ***
SALES       14       5     Z                1                              8       8
                                                                                      1   ***    SALES      ***
                                                                                      2   ***    1000'S     ***
** CLSAG00  TYPE 0  END OF ARRAY DEFINITION GLOSSARY.                                                          *****
NOV 05, 2001  16.56.48                                                                                PAGE     9
** CLSAA00  TYPE 0  DEFINITION COSTARRX IS BEING PROCESSED.                                                    *****
-------------------------------------------------------------------------------------------------------------------------------
   ARRAY   STMT        DELETE GLOSSARY RECORD RECORD RECORDS BUFFER   NUMBER   NUMBER                    SEQUENCE   LIST
   NAME    TYPE        ARRAY? TYPE     FORMAT SIZE   PER BLK  SIZE   OF ROWS  OF COLS                   73 TO 80   SEQ
-------------------------------------------------------------------------------------------------------------------------------
(COSTARRX) (AD)          (R)    (B)                                (    52) (     1)                              53

NOV 05, 2001  16.56.48                                                                                PAGE    10
                                  DETAILED GLOSSARY BY LOCATION FOR                                GLOSSARY PAGE     1
                                    ARRAY DEFINITION - COSTARRX
DATE CATALOGED - 11/05/01 16:56:48                          DATE LAST UPDATED - 11/05/01 16:56:48
CATALOGED BY -                                              LAST UPDATED BY -
EXPIRATION DATE -                                           DATE LAST USED -
NUMBER OF ROWS = 52                                         RECORD FORMAT = FIXED BLOCKED (DEFAULT)
NUMBER OF COLUMNS = 1                                       RECORD SIZE = 7492          (COMPUTED)
DATA CELL SIZE = 144                                        RECORDS PER BLOCK = 1       (DEFAULT)
ARRAY SIZE = 7492
-------------------------------------------------------------------------------------------------------------------------------
FIELD     FIELD    FIELD  FIELD  FIELD    DEC     ARRAY     EDIT CODES     EDIT    OUTPUT   LINE       REPORT FIELD
NAME      LOCATION LENGTH TYPE   RNDING   PLACES  KEY       ( ) ( ) ( )    LENGTH  WIDTH    NO  ***    HEADING    ***
-------------------------------------------------------------------------------------------------------------------------------
YEAR         1       4     C                        1                      4       4
-------------------------------------------------------------------------------------------------------------------------------
WEEKNO       1       2     C                                               2       2
COSTS1       9       5     Z                1                              8       8
SALES1      14       5     Z                1                              8       8
COSTS2      27       5     Z                1                              8       8
SALES2      32       5     Z                1                              8       8
COSTS3      45       5     Z                1                              8       8
SALES3      50       5     Z                1                              8       8
COSTS4      63       5     Z                1                              8       8
SALES4      68       5     Z                1                              8       8
COSTS5      81       5     Z                1                              8       8
SALES5      86       5     Z                1                              8       8
COSTS6      99       5     Z                1                              8       8
SALES6     104       5     Z                1                              8       8
COSTS7     117       5     Z                1                              8       8
SALES7     122       5     Z                1                              8       8
COSTS8     135       5     Z                1                              8       8
SALES8     140       5     Z                1                              8       8
** CLSAG00  TYPE 0  END OF ARRAY DEFINITION GLOSSARY.                                                          *****
NOV 05, 2001  16.56.48                                                                                PAGE    11
** CLSAA00  TYPE 0  DEFINITION FD4ARRAY IS BEING PROCESSED.                                                   *****
-------------------------------------------------------------------------------------------------------------------------------
   FILE    STMT    FILE    DELETE GLOSSARY RECORD RECORD RECORDS BUFFER MAPPED     MAPPING        MAPPED      SEQUENCE   LIST
   NAME    TYPE    IDENT   FILE   LEVEL    FORMAT SIZE   PER BLK  SIZE   FILE     INIT     TERM  RECSIZ      73 TO 80   SEQ
-------------------------------------------------------------------------------------------------------------------------------
(FD4ARRAY) (FD)            (R)    (B)      (K)                ( 7492)                                            72

NOV 05, 2001  16.56.48                                                                                PAGE    12
                                        FILE STRUCTURE DIAGRAM FOR                                DIAGRAM PAGE  1
                                        FILE DEFINITION - FD4ARRAY
RECORD FORMAT = KEY SEQUENCED VSAM                                  FILE IDENTIFICATION =
RECORD SIZE = 7492    BUFFER SIZE = 7492                            NUMBER OF SEGMENTS IN FILE = 2
```

Figure D-40 Preparation Job (Page 4 of 9)

```
                                                  +---------+
                                                  °SEG001L1 °
                                                  +---------+
                                                       °
                                                       °
                                                       °
                                                  +---------+
                                                  °SEG002L2 °
                                                  +---------+
NOV 05, 2001  16.56.48                                                                                    PAGE     13
                                        DETAILED GLOSSARY BY LOCATION FOR                                GLOSSARY PAGE      1
                                            FILE DEFINITION - FD4ARRAY
DATE CATALOGED - 11/05/01 16:56:48                                                DATE LAST UPDATED - 11/05/01 16:56:48
CATALOGED BY -                                                                    LAST UPDATED BY -
EXPIRATION DATE -                                                                 DATE LAST USED -
FILE IDENTIFICATION =                                                             RECORD FORMAT = KEY SEQUENCED VSAM
NUMBER OF SEGMENTS IN FILE = 2                                                    RECORD SIZE = 7492
NUMBER OF FIELDS IN FILE = 7                                                      BUFFER SIZE = 7492

                                         ********************************
                                         *    SEGMENT   1, LEVEL 1     *
                                         ********************************
SEGMENT OCCURS N TIMES = 1                                        KEY FIELD 1 = YEAR       TYPE = C   LENGTH = 4
SEGMENT SIZE = 4
NUMBER OF FIELDS IN SEGMENT = 1
-----------------------------------------------------------------------------------------------------------------------------
FIELD   FIELD   FIELD    FIELD   FIELD    DEC    CNT FIELD  EDIT CODES     EDIT    OUTPUT   LINE  DATA BASE NAME/
NAME    TYPE   LOCATION  LENGTH  RNDING  PLACES  FOR SGMT   ( ) ( ) ( )   LENGTH   WIDTH     NO   *** COLUMN HEADING ***
-----------------------------------------------------------------------------------------------------------------------------
YEAR      C        1       4                                                 4       4
                                                                                                1    ***      YEAR      ***
                                                                                                2    ***      NO.       ***
NOV 05, 2001  16.56.48                                                                                    PAGE     14
                                        DETAILED GLOSSARY BY LOCATION FOR                                GLOSSARY PAGE      2
                                            FILE DEFINITION - FD4ARRAY

                                         ********************************
                                         *    SEGMENT   2, LEVEL 2     *
                                         ********************************
SEGMENT OCCURS N TIMES = 416                                      KEY FIELD 1 = KEYSEG2    TYPE = C   LENGTH = 3
SEGMENT SIZE = 18
NUMBER OF FIELDS IN SEGMENT = 6
-----------------------------------------------------------------------------------------------------------------------------
FIELD   FIELD   FIELD    FIELD   FIELD    DEC    CNT FIELD  EDIT CODES     EDIT    OUTPUT   LINE  DATA BASE NAME/
NAME    TYPE   LOCATION  LENGTH  RNDING  PLACES  FOR SGMT   ( ) ( ) ( )   LENGTH   WIDTH     NO   *** COLUMN HEADING ***
-----------------------------------------------------------------------------------------------------------------------------
KEYSEG2   C        1       3                                                 3       3
WEEKNO    C        1       2                                                 2       4
                                                                                                1    ***      WEEK      ***
                                                                                                2    ***      NO.       ***
DAYNO     C        3       1                                                 1       3
                                                                                                1    ***      DAY       ***
                                                                                                2    ***      NO.       ***
HOURS     Z        4       5            1                                    8       8
                                                                                                1    ***     HOURS      ***
                                                                                                2    ***     WORKED     ***
COSTS     Z        9       5            1                                    8       8
                                                                                                1    ***     COSTS      ***
                                                                                                2    ***     1000'S     ***
SALES     Z       14       5            1                                    8       8
                                                                                                1    ***     SALES      ***
                                                                                                2    ***     1000'S     ***
** CLSGB00  TYPE 0  END OF FILE DEFINITION GLOSSARY.                                                                  *****
NOV 05, 2001  16.56.48                                                                                    PAGE     15
** MK4W204  TYPE 0  NUMBER OF MESSAGES PRINTED IS   19.                                                                *****
                   M4INPUT  -       91 RECORDS INPUT
                   M4LIST   -      419 RECORDS OUTPUT
                     15 TRACKS ASSIGNED TO M4LIB    --   13 TRACKS NOT FULL. LIBRARY DIRECTORY BLOCKING FACTOR IS  800.
** MK4W209  TYPE 0   924904 BYTES OF MAIN STORAGE UNUSED DURING DICTIONARY MAINTENANCE                                *****

DATA SET UTILITY - GENERATE                                                                    PAGE 0001
   GENERATE MAXFLDS=10
     RECORD FIELD=(50)

PROCESSING ENDED AT EOD
IDCAMS  SYSTEM SERVICES                                          TIME: 16:56:50       11/05/01     PAGE      1

  REPRO INFILE(SEQL) -
    OUTDATASET(BUILDER.R140.SALESAMT.R13#1)   /* <<< D... S.. N...   */
IDC0005I NUMBER OF RECORDS PROCESSED WAS 364
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0

                                          /* <<< S.. S... IDCAMS */
IDC0002I IDCAMS PROCESSING COMPLETE. MAXIMUM CONDITION CODE WAS 0
DATA SET UTILITY - GENERATE                                                                    PAGE 0001
```

Figure D-40 Preparation Job (Page 5 of 9)

```
   GENERATE MAXFLDS=10
     RECORD FIELD=(80)

PROCESSING ENDED AT EOD

NOV 05, 2001  16.56.53                                                                                    PAGE      1
                                          ******************************************
                                          * VISION:Builder   4400  (OS/390 - 14.0)  *
                                          *            COPYRIGHT (C) 2002           *
                                          *  COMPUTER ASSOCIATES INTERNATIONAL, INC. *
                                          ******************************************
VISION:Builder JCR1 14.0 110201              ENABLED FOR IBM LANGUAGE ENVIRONMENT                BUILD STAMP = 101306,12:51:24.
                                           COMLIB  MODIFICATIONS (PTFs, APARs) INSTALLED
                                        201,202,203,204,205,206,207,208,209,210,212,213,214,215,216
=============================================================================================================================
=                                                                                                                           =
=                    I N S T A L L A T I O N   P A R A M E T E R S   ( M 4 P A R A M S , M A R K L I B P )                  =
=                                                                                                                           =
=  SYSTEM DELIMITER:   #          PAGE HEIGHT:        66        M4LIST WIDTH:      132       DEF WIDTH OF PAGE:  0        =
=                                                                                                                           =
=  AUTO GRAND:         N          HEADING CHAR:       -         SUBTITLE REPEAT:   N         INVALID FIELD:      *        =
=                                                                                                                           =
=  MISSING FIELD:      -          NON-EDIT FIELD:     +         PERCENT CHAR:      %         LEFT SEPARATOR:     (        =
=                                                                                                                           =
=  RIGHT SEPARATOR:    )          SINGLE SEPARATOR:   ,         SOURCE SPACING:    1         PRINT MESSAGES:     Y        =
=                                                                                                                           =
=  CONSOLE MESSAGES:   N          M4REPO BLOCKSIZE:   4,096     INPUT I/O BUFFERS: 2         OUTPUT I/O BUFFERS: 1        =
=                                                                                                                           =
=  SNGL-STEP STORAGE:  8,192      SNGL-STEP SORTSIZE: 524,288   DIGIT SELECT CHAR: 9         ZERO SUPPRESS CHAR: Z        =
=                                                                                                                           =
=  CURRENCY CHAR:      $          PLUS CHAR:          +         MINUS CHAR:        -         CHECK PROTECT CHAR: *        =
=                                                                                                                           =
=  DECIMAL CHAR:       .          GROUPING CHAR:      ,         PRIMARY PLOT CHAR: X         SECONDARY PLOT CHAR:*        =
=                                                                                                                           =
=  FIT PLOT CHAR:      .          HORIZONTAL AXIS:    _         HORIZONTAL HASH:   |         VERTICAL AXIS:     |        =
=                                                                                                                           =
=  VERTICAL HASH:      -          MINUTES/HOUR:       60        SECONDS/MINUTE:    60        TIME DELIMITER:     HH:MM:SS =
=                                                                                                                           =
=  DATE FORMAT:      MM DD, YYYY  TODAY FORMAT+DELIM: MM/DD/YY  ISDATE DELIMITER:  YYYY-MM-DD  JULIAN DELIMITER:  YY.DDD    =
=                                                                                                                           =
=  SORT PROGRAM CODE:  2          MINCORE VALUE:      12 K      ALT M4LIST WIDTH:  264       ALT DEF W/OF PAGE:  0        =
=                                                                                                                           =
=  MAX LINES OF TRACE: 1,024      ITEM TRACKING:      0         SUPPRESS NDS REPT?: N        DEFAULT MAXGETMN:   1,024 K  =
=                                                                                                                           =
=  CONDITION CODE 1:   0          CONDITION CODE 2:   4         CONDITION CODE 3:  0         CONDITION CODE 4:   16       =
=                                                                                                                           =
=============================================================================================================================
NOV 05, 2001  16.56.53                                                                                    PAGE      2
-----------------------------------------------------------------------------------------------------------------------------
   RUN     STMT    FILE    O N S D U T R A B S D S L  RB  R S M R S          START          END            SEQUENCE   LIST
   NAME    TYPE    NAME    L E E I P R E U U R L C S  PL  J O O F S          SEARCH         SEARCH         73 TO 80   SEQ
                           D W Q R D N P D F T M N T  TK  T P P O R
-----------------------------------------------------------------------------------------------------------------------------
(LOADWHRS) (RC) (CARD    ) (S)         (S)       (T,G)         (R)                                      (ASLGEN'D)      1
    CONTROL NAME LOADWHRS TERM LISTGEN
    FILE MASTER INPUT CARD
    FILE SUBF1  ARRAYOUT
    FILE REPORT
    ARRAY A NAME COSTARR ; "A" will qualify the Array Field Names
    ;
    ; End Run Control
    ;  a fixed format instream File Definition will follow
    ;   the FD only applies to this application
    ;
    ;;ENDASL
-----------------------------------------------------------------------------------------------------------------------------
   RUN     STMT  LOGICAL  LBL  DTF/DD     KEY      KEY      KEY      QUALIFIER   USER I/O  ARRAY      PASSWORD/  SEQUENCE   LIST
   NAME    TYPE  FILENAME       NAME     FIELD-1  FIELD-2  FIELD-3  ICF CRD DBI  MODULE    QUAL       AUTH.-ID   73 TO 80   SEQ
-----------------------------------------------------------------------------------------------------------------------------
(LOADWHRS) (RF) (CARD    ) (S) (M4OLD    )                                                             (ASLGEN'D)      2
(LOADWHRS) (RF) (ARRAYOUT) (S) (M4SUBF1 )                                                             (ASLGEN'D)      3
(LOADWHRS) (RF) (COSTARR )                                                        (A)                 (ASLGEN'D)      4
NOV 05, 2001  16.56.53                                                                                    PAGE      3
** CLSAA00  TYPE 0  DEFINITION CARD     IS BEING PROCESSED.                                                          *****
-----------------------------------------------------------------------------------------------------------------------------
   FILE    STMT    FILE     DELETE GLOSSARY RECORD RECORD RECORDS BUFFER MAPPED      MAPPING          MAPPED      SEQUENCE   LIST
   NAME    TYPE    IDENT     FILE   LEVEL   FORMAT  SIZE  PER BLK  SIZE   FILE    INIT      TERM   RECSIZ      73 TO 80   SEQ
-----------------------------------------------------------------------------------------------------------------------------
(CARD    ) (FD)              (R)              (F)    (  80)                                                 (00441001)      5
-----------------------------------------------------------------------------------------------------------------------------
   FILE    STMT    FIELD    DEL SGMT LEV FIELD  FIELD FLD SEG FLD DEC CNT OCRS OUTP EDIT      DATA BASE NAME/          SEQUENCE   LIST
   NAME    TYPE    NAME     ETE NMBR NUM LOCATN LNGTH TYP KEY RND PLC SEG NTIM EDIT LNGH      COLUMN HEADING           73 TO 80   SEQ
-----------------------------------------------------------------------------------------------------------------------------
(CARD    ) (LO) (KEY      )   ( 1) (1) (   1) (  3) (C) (1)                                              (00442001)      6
(CARD    ) (LO) (ARRDATA  )   ( 1) (1) (   1) ( 18) (C)                                                 (00443001)      7
(CARD    ) (LO) (WEEKNO   )   ( 1) (1) (   1) (  2) (C)                                                 (00443001)      8
```

Figure D-40 Preparation Job (Page 6 of 9)

```
(CARD    ) (L0) (DAYNO   )    ( 1) (1) (   3) (  1) (C)                                                    (00443001)     9
(CARD    ) (L0) (HOURS   )    ( 1) (1) (   4) (  5) (Z)         (1)                                        (00443001)    10
(CARD    ) (L0) (COSTS   )    ( 1) (1) (   9) (  5) (Z)         (1)                                        (00443001)    11
(CARD    ) (L0) (SALES   )    ( 1) (1) (  14) (  5) (Z)         (1)                                        (00443001)    12
** CLSUN00  TYPE 1  SEGMENT   1 LEVEL 1 CONTAINS OVERLAPPING FIELDS.                                                      *****
                   ARRDATA  OVERLAPS KEY      WEEKNO   DAYNO    HOURS                                                      *****
                                     COSTS    SALES                                                                      *****
                   KEY      OVERLAPS WEEKNO   DAYNO                                                                      *****
NOV 05, 2001  16.56.53                                                                                          PAGE      4
                                          DETAILED GLOSSARY BY NAME FOR                                  GLOSSARY PAGE      1
                                            FILE DEFINITION - CARD
DATE CATALOGED -                                                                  DATE LAST UPDATED -
CATALOGED BY -                                                                    LAST UPDATED BY -
EXPIRATION DATE -                                                                 DATE LAST USED -
FILE IDENTIFICATION =                                                             RECORD FORMAT = FIXED UNBLOCKED
NUMBER OF SEGMENTS IN FILE = 1                                                    RECORD SIZE = 80
NUMBER OF FIELDS IN FILE = 7                                                      BLOCK SIZE = 80

                                         ********************************
                                         *    SEGMENT   1, LEVEL 1     *
                                         ********************************
SEGMENT OCCURS N TIMES = 1                                        KEY FIELD 1 = KEY        TYPE = C   LENGTH = 3
SEGMENT SIZE = 18
NUMBER OF FIELDS IN SEGMENT = 7
-----------------------------------------------------------------------------------------------------------------------------
FIELD   FIELD   FIELD    FIELD   FIELD    DEC    CNT FIELD  EDIT CODES     EDIT    OUTPUT   LINE  DATA BASE NAME/
NAME    TYPE   LOCATION  LENGTH  RNDING  PLACES  FOR SGMT   ( ) ( ) ( )   LENGTH   WIDTH     NO   *** COLUMN HEADING ***
-----------------------------------------------------------------------------------------------------------------------------
ARRDATA   C        1       18                                              18       18
COSTS     Z        9        5             1                                 8        8
DAYNO     C        3        1                                               1        1
HOURS     Z        4        5             1                                 8        8
KEY       C        1        3                                               3        3
SALES     Z       14        5             1                                 8        8
WEEKNO    C        1        2                                               2        2
** CLSGB00  TYPE 0  END OF FILE DEFINITION GLOSSARY.                                                                      *****
NOV 05, 2001  16.56.53                                                                                          PAGE      5


=============================================================================================================================
=                                                                                                                           =
=                                                  F I L E   S U M M A R Y                                                  =
=                                                                                                                           =
= JOBNAME: KREJO04X                          STEPNAME: PROC1                                                                =
=                                                                                                                           =
=  FILE NAME: M4OLD                          DSNAME: KREJO04.KREJO04X.JOB05838.D0000108.?                                   =
=  FILE NAME: M4SUBF1                        DSNAME: BUILDER.R140.ARR2001.R13#1                                             =
=  FILE NAME: M4LIB                          DSNAME: BUILDER.R140.M4LIB.R13#1                                               =
=                                                                                                                           =
=============================================================================================================================
NOV 05, 2001  16.56.53                                                                                          PAGE      6
                                                  ******************************
                                                  *  PROC NAME - MAIN          *
                                                  *  INPUT STREAM PROCEDURE    *
                                                  ******************************
-----------------------------------------------------------------------------------------------------------------------------
STMT REPORT                                 MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID           ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM       73 TO 80    SEQ
-----------------------------------------------------------------------------------------------------------------------------
(ER)                                                                                (N)                          (ASLGEN'D)    13
     ;;BEGASL
     ;
     ; Begin ASL coding again
     ;
     MAIN: PROC
     ;
     ; This application reads data (array data cell records)
     ; and loads it into an Array.  At EOF, the Entire Array is
     ; written to a subfile.  The purpose is to created an Array that
     ; contains Year 2002 Data for use in a subsequent example.
     ;
     ;
     ; Point to an Array Cell (Row/Column) based on a Week and Day number
     ; and move the data from the Master Record to the Array Cell
     ;
     IF LOCATE(ARRAY A ROW O.WEEKNO COLUMN O.DAYNO) THEN
       LET A.WEEKNO = O.WEEKNO
       LET A.DAYNO  = O.DAYNO
       LET A.HOURS  = O.HOURS
       LET A.COSTS  = O.COSTS
       LET A.SALES  = O.SALES
     ELSE
       CALL REPORT INPUTERR    ; Bad Week or Day Number
     END IF
     ;
```

Figure D-40 Preparation Job (Page 7 of 9)

```
    END PROC
    ;
-------------------------------------------------------------------------------------------------------------------------------
STMT SEQ  LOG CON .....OPERAND-A.....  OPER  .............OPERAND-B.............   ......RESULT......  PARTIAL FIELD  SEQUENCE   LIST
TYPE NO.  LEV CTR QLF FIELD   SEG-LVL  ATN   QLF FIELD   SEG-LVL  (OR CONSTANT)   QLF FIELD  SEG-LVL  STRT NUMB OPR  73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(PR) (100)          (A)               (LD)   (O,WEEKNO  ) 001-1                  (O,DAYNO   ) 001-1                (ASLGEN'D)    14
(PR) (101)                            (NS)     (108                          )                                     (ASLGEN'D)    15
(PR) (102)                            (R )   (O,WEEKNO  ) 001-1                  (A,WEEKNO  )                      (ASLGEN'D)    16
(PR) (103)                            (R )   (O,DAYNO   ) 001-1                  (A,DAYNO   )                      (ASLGEN'D)    17
(PR) (104)                            (R )   (O,HOURS   ) 001-1                  (A,HOURS   )                      (ASLGEN'D)    18
(PR) (105)                            (R )   (O,COSTS   ) 001-1                  (A,COSTS   )                      (ASLGEN'D)    19
(PR) (106)                            (R )   (O,SALES   ) 001-1                  (A,SALES   )                      (ASLGEN'D)    20
(PR) (107)                            (GO)     (END                          )                                     (ASLGEN'D)    21
(PR) (108)                            (GO)     (SUB INPUTERR                 )                                     (ASLGEN'D)    22
NOV 05, 2001  16.56.53                                                                                                 PAGE      7

                                                  ****************************
                                                  *  PROC NAME - ENXINPUT    *
                                                  *  INPUT STREAM PROCEDURE  *
                                                  ****************************

-------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                              MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID         ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM      73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(ER)                                                                                (E)                            (ASLGEN'D)    23
    ENXINPUT: PROC TYPE EOF
    ;
    ; All input has been read and moved to the Array
    ; Add the Year value (Array Key Field) and
    ; and Output the Entire Array (known as "A") to a Subfile              00450001
    ;                                                                     00451001
    LET A.YEAR = 2001                                                     00451001
    ;                                                                     00451001
    EXTRACT FILE ARRAYOUT ENTIRE A RECFM KEYVSAM                          00451001
    ;                                                                     00451001
    END PROC                                                              00451001
    ;                                                                     00451001
    ; List input data errors - bad week or day number
    ;                                                                     00451001
-------------------------------------------------------------------------------------------------------------------------------
STMT SEQ  LOG CON .....OPERAND-A.....  OPER  .............OPERAND-B.............   ......RESULT......  PARTIAL FIELD  SEQUENCE   LIST
TYPE NO.  LEV CTR QLF FIELD   SEG-LVL  ATN   QLF FIELD   SEG-LVL  (OR CONSTANT)   QLF FIELD  SEG-LVL  STRT NUMB OPR  73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(PR) (100)                            (R )   (D,2001                         ) (A,YEAR    )                      (ASLGEN'D)    24
(PR) (101)                            (GO)     (SUB SUBF___1                 )                                     (ASLGEN'D)    25
NOV 05, 2001  16.56.53                                                                                                 PAGE      8

                                                  ****************************
                                                  *  REQUEST NAME - SUBF___1 *
                                                  *  GENERATED REQUEST ____  *
                                                  ****************************

-------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                              MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID         ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM      73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(ER)                                                                                (S)                            (ASLGEN'D)    26
-------------------------------------------------------------------------------------------------------------------------------
 ST SUM V  8  PG  PG   I SP  MAXIMUM TB COLUMN  DT  PG  STRT LIN SL RH .......SUBFILE........ C E A S R S N      C  SEQUENCE   LIST
 TY RPT S LPI WID HGT  M FRM LPP PGS TO HEADING POS POS PAGE NUM UB PN ENT    NAME   BLK/ FMT H F G T S U D      O  73 TO 80    SEQ
 MP     P            G            LT TYP POS         NUM      ML TD REC          BUF      C C S L T M S      N
-------------------------------------------------------------------------------------------------------------------------------
(E1)                                                    (NR) (A) (ARRAYOUT)       (K)                            (ASLGEN'D)    27
NOV 05, 2001  16.56.53                                                                                                 PAGE      9

                                                  ****************************
                                                  *  PROC NAME - INPUTERR    *
                                                  *  INPUT STREAM PROCEDURE  *
                                                  ****************************

-------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                              MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID         ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM      73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(ER)                                                                                (S)                            (ASLGEN'D)    28
    INPUTERR: REPORT O.WEEKNO O.DAYNO O.HOURS O.COSTS,
                     O.SALES F.M4OLD F.ASTATUS,
                     TYPE SUBROUTINE
              TITLE 'DETAIL LIST OF OF BAD INPUT DATA'
              TITLE '- BAD WEEK OR DAY NUMBER -'
              END REPORT
-------------------------------------------------------------------------------------------------------------------------------
 ST SUM V  8  PG  PG   I SP  MAXIMUM TB COLUMN  DT  PG  STRT LIN SL RH .......SUBFILE........ C E A S R S N      C  SEQUENCE   LIST
 TY RPT S LPI WID HGT  M FRM LPP PGS TO HEADING POS POS PAGE NUM UB PN ENT    NAME   BLK/ FMT H F G T S U D      O  73 TO 80    SEQ
 MP     P            G            LT TYP POS         NUM      ML TD REC          BUF      C C S L T M S      N
-------------------------------------------------------------------------------------------------------------------------------
(E1)                                                                                                               (ASLGEN'D)    29
-------------------------------------------------------------------------------------------------------------------------------
         SP  Q              E  N  S  D  C  S  T  C  C  M  M  A  G  D Q PCT/RAT     P                PARTIAL   SEQUENCE   LIST
STMT SEQ  BF  L  FIELD  SEG-  N  P  E  E  T  U  O  U  N  A  I  V  M  E L  FIELD   RA- C     PICTURE     STRT NO  73 TO 80    SEQ
```

Figure D-40 Preparation Job (Page 8 of 9)

```
TYPE NO.  COL F  NAME      LVL  D  R  Q  S  L  B  T  M  T  X  N  G  R  C F  NAME    TIO T      EDIT      CHR CHR
-------------------------------------------------------------------------------------------------------------------------------
(R1)          (O,WEEKNO   ) 001-1                                                                          (ASLGEN'D)    30
(R1)          (O,DAYNO    ) 001-1                                                                          (ASLGEN'D)    31
(R1)          (O,HOURS    ) 001-1                                                                          (ASLGEN'D)    32
(R1)          (O,COSTS    ) 001-1                                                                          (ASLGEN'D)    33
(R1)          (O,SALES    ) 001-1                                                                          (ASLGEN'D)    34
(R1)          (F,M4OLD    )                                                                                (ASLGEN'D)    35
(R1)          (F,ASTATUS  )                                                                                (ASLGEN'D)    36
-------------------------------------------------------------------------------------------------------------------------------
STMT  SEQ                                                                                                   SEQUENCE   LIST
TYPE  NO.                                          TEXT                                                     73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(T1) (101)                 (DETAIL LIST OF OF BAD INPUT DATA#                      )                        (ASLGEN'D)    37
(T1) (102)                 (- BAD WEEK OR DAY NUMBER -#                            )                        (ASLGEN'D)    38
NOV 05, 2001  16.56.53                                                                                          PAGE    10
** MK4W204  TYPE 0  NUMBER OF MESSAGES PRINTED IS    6.                                                                  *****
                    M4OLD    -      416 RECORDS INPUT
                    M4REPO   -       13 RECORDS OUTPUT
                    M4SUBF1  -        1 RECORDS OUTPUT
                    M4INPUT  -       66 RECORDS INPUT
                    M4LIST   -      274 RECORDS OUTPUT
                      15 TRACKS ASSIGNED TO M4LIB    --   13 TRACKS NOT FULL. LIBRARY DIRECTORY BLOCKING FACTOR IS  800.
** MK4W209  TYPE 0   913600 BYTES OF MAIN STORAGE UNUSED DURING DECODING PHASE                                      *****
                     915216 BYTES OF MAIN STORAGE UNUSED DURING COMPILATION PHASE
                     667296 BYTES OF MAIN STORAGE UNUSED DURING PROCESSING PHASE
** MK4ED02  TYPE 0  REQUEST-INPUTERR  REPORT-1  REQUESTOR ID-                                                       *****
                                          DETAIL LIST OF OF BAD INPUT DATA                                  PAGE 1
                                            - BAD WEEK OR DAY NUMBER -
-------------------------------------------------------------------------------------------------------------------------------
                                       M4OLD       ASTATUS
-------------------------------------------------------------------------------------------------------------------------------
   -  -        -        -        -          -      -
** MK4E203  TYPE 1  NO DATA SELECTED FOR THIS REPORT.                                                               *****
** MK4FT06  TYPE 1        7 FIELDS WERE PRINTED AS INVALID, MISSING OR OVERFLOW.                                    *****
** MK4FT03  TYPE 0  END OF REPORT.                                                                                  *****
** MK4W204  TYPE 0  NUMBER OF MESSAGES PRINTED IS    4.                                                             *****
                    M4REPO   -       12 RECORDS INPUT
                    M4LIST   -       12 RECORDS OUTPUT
** MK4W209  TYPE 0   992802 BYTES OF MAIN STORAGE UNUSED DURING REPORTING PHASE                                     *****
```

Figure D-40 Preparation Job (Page 9 of 9)

# Examples Jobs

## Array Examples Application ARRAYAPP: Banner Page

```
NOV 06, 2001  12.31.38                                                                                          PAGE      1
                                        ******************************************
                                        *  VISION:Builder   4400  (OS/390 - 14.0)  *
                                        *            COPYRIGHT (C) 2002            *
                                        *  COMPUTER ASSOCIATES INTERNATIONAL, INC. *
                                        ******************************************
VISION:Builder JCR1 14.0 110201              ENABLED FOR IBM LANGUAGE ENVIRONMENT                BUILD STAMP = 101306,12:51:24.
                                           COMLIB  MODIFICATIONS (PTFs, APARs) INSTALLED
                                     201,202,203,204,205,206,207,208,209,210,212,213,214,215,216
=================================================================================================================================
=                                                                                                                               =
=                    I N S T A L L A T I O N    P A R A M E T E R S    ( M 4 P A R A M S , M A R K L I B P )                    =
=                                                                                                                               =
=  SYSTEM DELIMITER:   #          PAGE HEIGHT:       66         M4LIST WIDTH:      132        DEF WIDTH OF PAGE:  0          =
=                                                                                                                               =
=  AUTO GRAND:         N          HEADING CHAR:      -          SUBTITLE REPEAT:   N          INVALID FIELD:      *          =
=                                                                                                                               =
=  MISSING FIELD:      -          NON-EDIT FIELD:    +          PERCENT CHAR:      %          LEFT SEPARATOR:     (          =
=                                                                                                                               =
=  RIGHT SEPARATOR:    )          SINGLE SEPARATOR:  ,          SOURCE SPACING:    1          PRINT MESSAGES:     Y          =
=                                                                                                                               =
=  CONSOLE MESSAGES:   N          M4REPO BLOCKSIZE:  4,096      INPUT I/O BUFFERS: 2          OUTPUT I/O BUFFERS: 1          =
=                                                                                                                               =
=  SNGL-STEP STORAGE:  8,192      SNGL-STEP SORTSIZE: 524,288   DIGIT SELECT CHAR: 9          ZERO SUPPRESS CHAR: Z          =
=                                                                                                                               =
=  CURRENCY CHAR:      $          PLUS CHAR:         +          MINUS CHAR:        -          CHECK PROTECT CHAR: *          =
=                                                                                                                               =
=  DECIMAL CHAR:       .          GROUPING CHAR:     ,          PRIMARY PLOT CHAR: X          SECONDARY PLOT CHAR:*          =
=                                                                                                                               =
=  FIT PLOT CHAR:      .          HORIZONTAL AXIS:   _          HORIZONTAL HASH:   |          VERTICAL AXIS:      |          =
=                                                                                                                               =
=  VERTICAL HASH:      -          MINUTES/HOUR:      60         SECONDS/MINUTE:    60         TIME DELIMITER:     HH:MM:SS   =
=                                                                                                                               =
```

Figure D-41 Array Examples Application ARRAYAPP: Banner Page (Page 1 of 2)

```
=  DATE FORMAT:      MMM DD, YYYY  TODAY FORMAT+DELIM: MM/DD/YY    ISDATE DELIMITER:   YYYY-MM-DD  JULIAN DELIMITER:   YY.DDD     =
=                                                                                                                                =
=  SORT PROGRAM CODE:  2           MINCORE VALUE:      12 K        ALT M4LIST WIDTH:   264         ALT DEF W/OF PAGE:  0          =
=                                                                                                                                =
=  MAX LINES OF TRACE: 1,024       ITEM TRACKING:      0           SUPPRESS NDS REPT?: N           DEFAULT MAXGETMN:   1,024 K    =
=                                                                                                                                =
=  CONDITION CODE 1:   0           CONDITION CODE 2:   4           CONDITION CODE 3:   0           CONDITION CODE 4:   16         =
=                                                                                                                                =
==================================================================================================================================
```

Figure D-41 Array Examples Application ARRAYAPP: Banner Page (Page 2 of 2)

## Array Examples Application ARRAYAPP: Run Control Commands – Part 1 of 10

```
==================================================================================================================================
NOV 06, 2001  12.31.38                                                                                                  PAGE      2
----------------------------------------------------------------------------------------------------------------------------------
   RUN     STMT    FILE    O N S D U T R A B S D S L  RB  R S M R S            START           END              SEQUENCE   LIST
   NAME    TYPE    NAME    L E E I P R E U U R L C S  PL  J O O F S            SEARCH          SEARCH           73 TO 80    SEQ
                           D W Q R D N P D F T M N T  TK  T P P O R
----------------------------------------------------------------------------------------------------------------------------------
(ARRAYAPP) (RC) (WORKHRS ) (S) (U)      (S)       (T,G)          (R)                                                (ASLGEN'D)     1
     CONTROL NAME ARRAYAPP TERM LISTGEN
     FILE MASTER INPUT WORKHRS KEYS NONE
     FILE CORD1 SALEAMTS DIRECT BY T.INDEXKEY
     FILE CORD2 COSTARR  DIRECT BY T.LASTYEAR ARRAY ; External Array Input
     FILE SUBF1 ARRAYOUT                            ; Output of an Array
     FILE REPORT
     ARRAY A NAME COSTARR                  ; the Primary Array
     ARRAY B NAME COSTARRX OVERDEFINES A   ; the Overdefined Array
     ;
----------------------------------------------------------------------------------------------------------------------------------
   RUN     STMT  LOGICAL  LBL  DTF/DD      KEY       KEY       KEY      QUALIFIER   USER I/O  ARRAY       PASSWORD/  SEQUENCE   LIST
   NAME    TYPE  FILENAME       NAME     FIELD-1   FIELD-2   FIELD-3   ICF CRD DBI   MODULE   QUAL        AUTH.-ID   73 TO 80    SEQ
----------------------------------------------------------------------------------------------------------------------------------
(ARRAYAPP) (RF) (WORKHRS ) (S) (M4OLD   )                                                                          (ASLGEN'D)     2
(ARRAYAPP) (RF) (SALEAMTS) (S) (M4CORD1 ) (INDEXKEY)                    (T)                                        (ASLGEN'D)     3
(ARRAYAPP) (RF) (COSTARR ) (S) (M4CORD2 ) (LASTYEAR)                    (T)                    (2)                 (ASLGEN'D)     4
(ARRAYAPP) (RF) (ARRAYOUT) (S) (M4SUBF1 )                                                                          (ASLGEN'D)     5
(ARRAYAPP) (RF) (COSTARR )                                                                    (A)                  (ASLGEN'D)     6
(ARRAYAPP) (RF) (COSTARRX)                                                                    (B)                  (ASLGEN'D)     7
----------------------------------------------------------------------------------------------------------------------------------
   RUN     STMT          PRIMARY QUALIFIER            SECONDARY QUALIFIER                                          SEQUENCE   LIST
   NAME    TYPE                                                                                                      73 TO 80    SEQ
----------------------------------------------------------------------------------------------------------------------------------
(ARRAYAPP) (RA)                (A)                            (B)                                                  (ASLGEN'D)     8
```

Figure D-42 Array Examples Application ARRAYAPP: Run Control Commands Part 1

## Array Examples Application ARRAYAPP: MAIN Procedure Coding – Part 2 of 10

```
==================================================================================================================================
=                                                                                                                                =
=                                                  F I L E   S U M M A R Y                                                       =
=                                                                                                                                =
= JOBNAME: KREJO04X                          STEPNAME: F319                                                                      =
=                                                                                                                                =
=  FILE NAME: M4OLD                          DSNAME: BUILDER.R140.WORKHRS.R13#1                                                  =
=  FILE NAME: M4CORD1                        DSNAME: BUILDER.R140.SALESAMT.R13#1                                                 =
=  FILE NAME: M4CORD2                        DSNAME: BUILDER.R140.ARR2001.R13#1                                                  =
=  FILE NAME: M4SUBF1                        DSNAME: BUILDER.R140.ARR2002.R13#1                                                  =
=  FILE NAME: M4LIB                          DSNAME: BUILDER.R140.M4LIB.R13#1                                                    =
=                                                                                                                                =
==================================================================================================================================
NOV 06, 2001  12.31.38                                                                                                  PAGE      3
                                                  ****************************
                                                  *  PROC NAME - MAIN        *
                                                  *  INPUT STREAM PROCEDURE  *
                                                  ****************************
----------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                                MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID          ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM       73 TO 80    SEQ
----------------------------------------------------------------------------------------------------------------------------------
(ER)                                                                                (N)                            (ASLGEN'D)     9
```

Figure D-43 Array Examples Application ARRAYAPP: MAIN Procedure Coding Part 2 (Page 1 of 2)

```
MAIN: PROC
;
; For each MASTER Work Hours Record Input
;
; Load the Internal Array
; with hours by week and day numbers
;
IF A.YEAR = '    '      ; set year in array rcd key field for
   LET A.YEAR = O.YEAR  ; we output the Array to a subfile
END IF
;
LOCATE ARRAY A ROW O.WEEKNO COLUMN O.DAYNO
;
;  (the week and day number values reference a specific
;   row & column {data cell}, no looping will occur.)
;
;     If the Week and Day Values are outside the dimensions of the
;     Array a Data Cell LOCATE Error occurs - test the ASTATUS Flag
;
IF F.ASTATUS = '    ' THEN              ; successful Locate
   LET A.HOURS = A.HOURS + O.HRSWRKD     ; move data into Array
   LET A.WEEKNO = O.WEEKNO
   LET A.DAYNO  = O.DAYNO
ELSE                                    ; unsuccessful Locate
   CALL REPORT HRSERR                   ; bad week or day value
END IF
;
END PROC
;
```

```
STMT SEQ  LOG CON .....OPERAND-A.....  OPER  ............OPERAND-B.............   ......RESULT......  PARTIAL FIELD  SEQUENCE   LIST
TYPE NO.  LEV CTR QLF FIELD   SEG-LVL  ATN   QLF FIELD   SEG-LVL  (OR CONSTANT)   QLF FIELD  SEG-LVL  STRT NUMB OPR  73 TO 80    SEQ
-----------------------------------------------------------------------------------------------------------------------------------
(PR) (100)         (A,YEAR    )        (EQ)   (C)                                                                  (ASLGEN'D)    10
(PR) (101)                             (NS)     (103                          )                                    (ASLGEN'D)    11
(PR) (102)                             (R )   (O,YEAR    ) 001-1                (A,YEAR    )                      (ASLGEN'D)    12
(PR) (103)         (A)                 (LD)   (O,WEEKNO  ) 001-1                (O,DAYNO   ) 001-1                (ASLGEN'D)    13
(PR) (104)         (F,ASTATUS )        (EQ)   (C)                                                                  (ASLGEN'D)    14
(PR) (105)                             (NS)     (110                          )                                    (ASLGEN'D)    15
(PR) (106)         (A,HOURS   )        (+ )   (O,HRSWRKD ) 001-1                (A,HOURS   )                      (ASLGEN'D)    16
(PR) (107)                             (R )   (O,WEEKNO  ) 001-1                (A,WEEKNO  )                      (ASLGEN'D)    17
(PR) (108)                             (R )   (O,DAYNO   ) 001-1                (A,DAYNO   )                      (ASLGEN'D)    18
(PR) (109)                             (GO)     (END                          )                                    (ASLGEN'D)    19
(PR) (110)                             (GO)     (SUB HRSERR                   )                                    (ASLGEN'D)    20
```

Figure D-43 Array Examples Application ARRAYAPP: MAIN Procedure Coding Part 2 (Page 2 of 2)

## Array Examples Application ARRAYAPP: EOFMAIN Procedure Coding – Part 3 of 10

```
NOV 06, 2001  12.31.38                                                                                                  PAGE      4
                                                  ******************************
                                                  *  PROC NAME - EOFMAIN      *
                                                  *  INPUT STREAM PROCEDURE   *
                                                  ******************************
-----------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                                MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID          ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM       73 TO 80    SEQ
-----------------------------------------------------------------------------------------------------------------------------------
(ER)                                                                                (E)                              (ASLGEN'D)    21

    EOFMAIN: PROC TYPE EOF
    ;
    ; EOF Procedure - after all MASTER Work Hours Records are processed.
    ;
    CALL PROCEDURE FILLINA  ; Calc cost, retrieve sales,
                            ; and put into the internal Array
                            ;
    CALL REPORT COSTRPT1    ; Produce Report 1 (std report)
                            ; cost versus sales
                            ;
    CALL REPORT COSTRPT2    ; Produce Report 2 (customized)
                            ; cost versus sales across page
                            ; using overdefined array def
                            ;
    CALL SUBFILE ARROUT     ; Write the internal Array to a Subfile
                            ;
    CALL PROCEDURE ARRIN    ; Read an external Array (Cord 2) for
                            ; Last Year's Data and Report the Information
```

Figure D-44 Array Examples Application ARRAYAPP: EOFMAIN Procedure Coding Part 3 (Page 1 of 2)

```
     ;
     END PROC
     ;
-------------------------------------------------------------------------------------------------------------------------------
STMT SEQ  LOG CON .....OPERAND-A.....  OPER  ............OPERAND-B.............   ......RESULT......  PARTIAL FIELD  SEQUENCE   LIST
TYPE NO.  LEV CTR QLF FIELD   SEG-LVL  ATN   QLF FIELD   SEG-LVL  (OR CONSTANT)   QLF FIELD  SEG-LVL  STRT NUMB OPR  73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(PR) (100)                              (GO)     (SUB FILLINA                  )                                  (ASLGEN'D)    22
(PR) (101)                              (GO)     (SUB COSTRPT1                 )                                  (ASLGEN'D)    23
(PR) (102)                              (GO)     (SUB COSTRPT2                 )                                  (ASLGEN'D)    24
(PR) (103)                              (GO)     (SUB ARROUT                   )                                  (ASLGEN'D)    25
(PR) (104)                              (GO)     (SUB ARRIN                    )                                  (ASLGEN'D)    26
```

Figure D-44 Array Examples Application ARRAYAPP: EOFMAIN Procedure Coding Part 3 (Page 2 of 2)

## Array Examples Application ARRAYAPP: FILLINA Procedure – Part 4 of 10

```
NOV 06, 2001  12.31.38                                                                                                  PAGE      5
                                                  ******************************
                                                  *  PROC NAME - FILLINA      *
                                                  *  INPUT STREAM PROCEDURE   *
                                                  ******************************
-------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                                  MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID            ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM       73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(ER)                                                                                  (S)                              (ASLGEN'D)    27
     FILLINA: PROC TYPE SUBROUTINE
     ;
     ; Finish building the internal Array Data
     ; Calculate the Cost Amounts
     ; Retrieve Sales Data from the ICF (DIRECT) Cord File
     ;  and move it into the Array
     ; Accumulate the Weekly Sales for each row in the Array
     ;  and store the Amount into the Array
     ;
     INDEXKEY: FIELD C 7    ; Temp Field for DIRECT Cord1 Read
     ;
     ; Calculate Cost in 1000's for each column (day)
     ; (this starts an automatic loop thru the entire Array)
     ;
     LET A.COSTS = A.HOURS * 7.67 / 1000 WITH ROUNDING
     ;
     ; When you get to column 8 in each row, use it to store
     ; the Weekly Accumulated Totals from each column in
     ; the current row.
     ;
     IF F.COLUMN = 8 THEN            ; Column 8 processing
        LET A.COSTS = T.WKCOSTS      ; Store the accum'd values
        LET A.SALES = T.WKSALES      ; from columns 1 to 7 in the row
        LET A.WEEKNO = T.WEEKNO
        LET A.DAYNO  = '8'
        LET T.WKCOSTS = 0            ; Clear accum'd values
        LET T.WKSALES = 0
     ;
     ; If not column 8, it will be column 1 thru 7
     ;
     ; Build the DIRECT Key Field to read the Sales Cord File
     ; Retrieve a Sales Record and move the Sales Amount into the Array
     ;
     ; Accumulate the Costs and Sales for Weekly Totals Column
     ;
     ELSE                            ; Column 1 to 7 processing
     ;
       COMBINE A.YEAR A.WEEKNO A.DAYNO STORE T.INDEXKEY   ; build DIRECT Key
     ;
       LET A.SALES   = 1.SALES                  ; <- This causes DIRECT Read
       LET T.WKCOSTS = T.WKCOSTS + A.COSTS      ; Accum Totals
       LET T.WKSALES = T.WKSALES + A.SALES
       LET T.WEEKNO  = A.WEEKNO
     ;
     END IF
     ;       Returns when all rows have been processed (Automatic Loop Ends)
     ;
     END PROC
     ;
NOV 06, 2001  12.31.38                                                                                                  PAGE      6
-------------------------------------------------------------------------------------------------------------------------------
STMT   FIELD    FIELD FLD DEC OUTPT EDIT      INITIAL          LINE 1 OF        LINE 2 OF                          SEQUENCE   LIST
```

Figure D-45 Array Examples Application ARRAYAPP: FILLINA Procedure Part 4 (Page 1 of 2)

```
TYPE   NAME      LNGTH TYP PLC EDIT  LGTH       VALUE        COLUMN HEADING   COLUMN HEADING                          73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(TF) (INDEXKEY) (  7) (C)                                                                                          (ASLGEN'D)    28
(TF) (TEMP___7) ( 10) (P) (2)                                                                                      (ASLGEN'D)    29
-------------------------------------------------------------------------------------------------------------------------------
STMT SEQ  LOG CON .....OPERAND-A.....  OPER  .............OPERAND-B.............  ......RESULT......  PARTIAL FIELD  SEQUENCE   LIST
TYPE NO.  LEV CTR QLF FIELD   SEG-LVL  ATN   QLF FIELD   SEG-LVL  (OR CONSTANT)  QLF FIELD  SEG-LVL  STRT NUMB OPR  73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(PR) (100)         (A,HOURS   )       (*R)   (D,7.67                        ) (T,TEMP___7)                        (ASLGEN'D)    30
(PR) (101)         (T,TEMP___7)       (/R)   (D,1000                        ) (A,COSTS   )                        (ASLGEN'D)    31
(PR) (102)         (F,COLUMN  )       (EQ)   (D,8                           )                                     (ASLGEN'D)    32
(PR) (103)                            (NS)     (111                         )                                     (ASLGEN'D)    33
(PR) (104)                            (R )   (T,WKCOSTS )                      (A,COSTS   )                        (ASLGEN'D)    34
** MK4DI02  TYPE 0  DEFINITION OF TEMPORARY FIELD WKCOSTS  - FORMAT Z  LENGTH   5  DECIMALS 1 EDIT LENGTH   8 .                *****
(PR) (105)                            (R )   (T,WKSALES )                      (A,SALES   )                        (ASLGEN'D)    35
** MK4DI02  TYPE 0  DEFINITION OF TEMPORARY FIELD WKSALES  - FORMAT Z  LENGTH   5  DECIMALS 1 EDIT LENGTH   8 .                *****
(PR) (106)                            (R )   (T,WEEKNO  )                      (A,WEEKNO  )                        (ASLGEN'D)    36
** MK4DI02  TYPE 0  DEFINITION OF TEMPORARY FIELD WEEKNO   - FORMAT C  LENGTH   2  DECIMALS 0 EDIT LENGTH   2 .                *****
(PR) (107)                            (R )   (C,8                           ) (A,DAYNO   )                        (ASLGEN'D)    37
(PR) (108)                            (R )   (D,0                           ) (T,WKCOSTS )                        (ASLGEN'D)    38
(PR) (109)                            (R )   (D,0                           ) (T,WKSALES )                        (ASLGEN'D)    39
(PR) (110)                            (GO)     (END                         )                                     (ASLGEN'D)    40
(PR) (111)         (A,YEAR    )       (CO)   (A,WEEKNO  )                      (T,INDEXKEY)                        (ASLGEN'D)    41
(PR) (112)         (T,INDEXKEY)       (CO)   (A,DAYNO   )                      (T,INDEXKEY)                        (ASLGEN'D)    42
(PR) (113)                            (R )   (1,SALES   ) 001-1                (A,SALES   )                        (ASLGEN'D)    43
(PR) (114)         (T,WKCOSTS )       (+ )   (A,COSTS   )                      (T,WKCOSTS )                        (ASLGEN'D)    44
(PR) (115)         (T,WKSALES )       (+ )   (A,SALES   )                      (T,WKSALES )                        (ASLGEN'D)    45
(PR) (116)                            (R )   (A,WEEKNO  )                      (T,WEEKNO  )                        (ASLGEN'D)    46
```

Figure D-45 Array Examples Application ARRAYAPP: FILLINA Procedure Part 4 (Page 2 of 2)

## Array Examples Application ARRAYAPP: COSTRPT1 Report Procedure – Part 5 of 10

```
NOV 06, 2001  12.31.38                                                                                          PAGE      7

                                          ****************************
                                          *  PROC NAME - COSTRPT1     *
                                          *  INPUT STREAM PROCEDURE   *
                                          ****************************

-------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                              MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID        ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM       73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(ER) (TODAYX)                                                                    (S)                              (ASLGEN'D)    47

    COSTRPT1: REPORT,
    ;
    ; Report Costs/Sales by Day and Week
    ;
                A.WEEKNO A.DAYNO A.COSTS A.SALES,
          TYPE SUBROUTINE
          FORMAT WIDTH 80 DATEFMT TODAYX
          TITLE 'COST AND SALES BY DAY AND WEEK'
          TITLE 'AMOUNTS SHOWN IN 1000 OF DOLLARS'
          END REPORT
    ;
-------------------------------------------------------------------------------------------------------------------------------
 ST SUM V  8  PG  PG   I SP  MAXIMUM TB COLUMN  DT  PG  STRT LIN SL RH .......SUBFILE........ C E A S R S N      C  SEQUENCE   LIST
 TY RPT S LPI WID HGT  M FRM LPP PGS TO HEADING POS POS PAGE NUM UB PN ENT    NAME   BLK/ FMT H F G T S U D      O  73 TO 80    SEQ
 MP     P            G            LT TYP POS         NUM      ML TD REC           BUF      C C S L T M S      N
-------------------------------------------------------------------------------------------------------------------------------
(E1)        (080)                                                                                               (ASLGEN'D)    48
-------------------------------------------------------------------------------------------------------------------------------
         SP  Q              E  N  S  D  C  S  T  C  C  M  M  A  G  D Q PCT/RAT     P                PARTIAL   SEQUENCE   LIST
STMT SEQ BF  L  FIELD  SEG-  N  P  E  E  T  U  O  U  N  A  I  V  M  E L  FIELD  RA- C      PICTURE    STRT NO   73 TO 80    SEQ
TYPE NO. COL F  NAME     LVL  D  R  Q  S  L  B  T  M  T  X  N  G  R  C F  NAME   TIO T       EDIT      CHR CHR
-------------------------------------------------------------------------------------------------------------------------------
(R1)        (A,WEEKNO  )                                                                                        (ASLGEN'D)    49
(R1)        (A,DAYNO   )                                                                                        (ASLGEN'D)    50
(R1)        (A,COSTS   )                                                                                        (ASLGEN'D)    51
(R1)        (A,SALES   )                                                                                        (ASLGEN'D)    52
-------------------------------------------------------------------------------------------------------------------------------
STMT  SEQ                                                                                                       SEQUENCE   LIST
TYPE  NO.                                        TEXT                                                           73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(T1) (101)              (COST AND SALES BY DAY AND WEEK#                        )                               (ASLGEN'D)    53
(T1) (102)              (AMOUNTS SHOWN IN 1000 OF DOLLARS#                      )                               (ASLGEN'D)    54
```

Figure D-46 Array Examples Application ARRAYAPP: COSTRPT1 Report Procedure Part 5

## Array Examples Application ARRAYAPP: COSTRPT2 Report Procedure – Part 6 of 10

```
NOV 06, 2001  12.31.38                                                                                                PAGE      8
                                                ****************************
                                                *  PROC NAME - COSTRPT2    *
                                                *  INPUT STREAM PROCEDURE  *
                                                ****************************
-------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                              MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE         REQUESTOR ID        ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM      73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(ER) (TODAY )                                                                  (S)                            (ASLGEN'D)     55

    COSTRPT2: REPORT,
    ;
    ; Report Costs/Sales by Day and Week
    ; Columnize the Days Across the Page using the Overdefined Array
    ;
                  B.WEEKNO,
                  B.COSTS1 B.SALES1,
                  B.COSTS2 B.SALES2,
                  B.COSTS3 B.SALES3,
                  B.COSTS4 B.SALES4,
                  B.COSTS5 B.SALES5,
                  B.COSTS6 B.SALES6,
                  B.COSTS7 B.SALES7,
                  B.COSTS8 B.SALES8,
            TYPE SUBROUTINE
            FORMAT SUMMARYLABELS SUPPRESS
            TITLE 'REPORT OF COSTS AND SALES BY DAY WITHIN WEEK ',
                 '- AMOUNTS IN $1000'
            ITEM B.WEEKNO SPACES 1
            ITEM B.COSTS1 B.SALES1,
                 B.COSTS2 B.SALES2,
                 B.COSTS3 B.SALES3,
                 B.COSTS4 B.SALES4,
                 B.COSTS5 B.SALES5,
                 B.COSTS6 B.SALES6,
                 B.COSTS7 B.SALES7,
                 B.COSTS8 B.SALES8,
              PICTURE P'ZZZ9.9'
            TOTAL B.COSTS1 B.SALES1,
                  B.COSTS2 B.SALES2,
                  B.COSTS3 B.SALES3,
                  B.COSTS4 B.SALES4,
                  B.COSTS5 B.SALES5,
                  B.COSTS6 B.SALES6,
                  B.COSTS7 B.SALES7,
                  B.COSTS8 B.SALES8,
              AT LEVEL G
            SECTION COLUMNHEADING
            LINE 'WEEK   ....MONDAY...   ...TUESDAY...   ..WEDNESDAY..',
                 '   ...THURDAY...   ....FRIDAY...   ...SATURDAY..',
                 '   ....SUNDAY...   ..WEEK TOTAL.'
            LINE ' NO.   COSTS   SALES   COSTS   SALES   COSTS   SALES',
                 '   COSTS   SALES   COSTS   SALES   COSTS   SALES',
                 '   COSTS   SALES   COSTS   SALES'
            END SECTION
            END REPORT
    ;
NOV 06, 2001  12.31.38                                                                                                PAGE      9
-------------------------------------------------------------------------------------------------------------------------------
 ST SUM V  8  PG  PG   I SP  MAXIMUM TB COLUMN   DT  PG  STRT LIN SL RH .......SUBFILE..... C E A S R S N      C  SEQUENCE   LIST
 TY RPT S LPI WID HGT  M FRM LPP PGS TO HEADING POS POS PAGE NUM UB PN ENT    NAME   BLK/ FMT H F G T S U D      O  73 TO 80    SEQ
 MP     P            G            LT TYP POS         NUM      ML TD REC           BUF      C C S L T M S      N
-------------------------------------------------------------------------------------------------------------------------------
(E1)                                                          (X)                                             (ASLGEN'D)     56
-------------------------------------------------------------------------------------------------------------------------------
         SP  Q              E  N  S  D  C  S  T  C  C  M  M  A  G  D Q PCT/RAT     P                PARTIAL   SEQUENCE   LIST
STMT SEQ  BF  L  FIELD  SEG-  N  P  E  E  T  U  O  U  N  A  I  V  M  E L  FIELD  RA- C     PICTURE    STRT NO   73 TO 80    SEQ
TYPE NO.  COL F  NAME     LVL  D  R  Q  S  L  B  T  M  T  X  N  G  R  C F  NAME   TIO T      EDIT      CHR CHR
-------------------------------------------------------------------------------------------------------------------------------
(R1)      ( 1) (B,WEEKNO  )                                                                              (ASLGEN'D)     57
(R1)           (B,COSTS1  )                        (G)                                                   (ASLGEN'D)     58
** MK4EB09  TYPE 0  THE EXTENDED EDIT PATTERN FOR FIELD B,COSTS1   WILL BE USED - ZZZ9.9                  .              *****
(R1)           (B,SALES1  )                        (G)                                                   (ASLGEN'D)     59
** MK4EB09  TYPE 0  THE EXTENDED EDIT PATTERN FOR FIELD B,SALES1   WILL BE USED - ZZZ9.9                  .              *****
(R1)           (B,COSTS2  )                        (G)                                                   (ASLGEN'D)     60
** MK4EB09  TYPE 0  THE EXTENDED EDIT PATTERN FOR FIELD B,COSTS2   WILL BE USED - ZZZ9.9                  .              *****
(R1)           (B,SALES2  )                        (G)                                                   (ASLGEN'D)     61
** MK4EB09  TYPE 0  THE EXTENDED EDIT PATTERN FOR FIELD B,SALES2   WILL BE USED - ZZZ9.9                  .              *****
(R1)           (B,COSTS3  )                        (G)                                                   (ASLGEN'D)     62
** MK4EB09  TYPE 0  THE EXTENDED EDIT PATTERN FOR FIELD B,COSTS3   WILL BE USED - ZZZ9.9                  .              *****
```

Figure D-47 Array Examples Application ARRAYAPP: COSTRPT2 Report Procedure Part 6 (Page 1 of 2)

```
(R1)         (B,SALES3  )                        (G)                                                                  (ASLGEN'D)    63
** MK4EB09  TYPE 0  THE EXTENDED EDIT PATTERN FOR FIELD B,SALES3   WILL BE USED - ZZZ9.9                    .          *****
(R1)         (B,COSTS4  )                        (G)                                                                  (ASLGEN'D)    64
** MK4EB09  TYPE 0  THE EXTENDED EDIT PATTERN FOR FIELD B,COSTS4   WILL BE USED - ZZZ9.9                    .          *****
(R1)         (B,SALES4  )                        (G)                                                                  (ASLGEN'D)    65
** MK4EB09  TYPE 0  THE EXTENDED EDIT PATTERN FOR FIELD B,SALES4   WILL BE USED - ZZZ9.9                    .          *****
(R1)         (B,COSTS5  )                        (G)                                                                  (ASLGEN'D)    66
** MK4EB09  TYPE 0  THE EXTENDED EDIT PATTERN FOR FIELD B,COSTS5   WILL BE USED - ZZZ9.9                    .          *****
(R1)         (B,SALES5  )                        (G)                                                                  (ASLGEN'D)    67
** MK4EB09  TYPE 0  THE EXTENDED EDIT PATTERN FOR FIELD B,SALES5   WILL BE USED - ZZZ9.9                    .          *****
(R1)         (B,COSTS6  )                        (G)                                                                  (ASLGEN'D)    68
** MK4EB09  TYPE 0  THE EXTENDED EDIT PATTERN FOR FIELD B,COSTS6   WILL BE USED - ZZZ9.9                    .          *****
(R1)         (B,SALES6  )                        (G)                                                                  (ASLGEN'D)    69
** MK4EB09  TYPE 0  THE EXTENDED EDIT PATTERN FOR FIELD B,SALES6   WILL BE USED - ZZZ9.9                    .          *****
(R1)         (B,COSTS7  )                        (G)                                                                  (ASLGEN'D)    70
** MK4EB09  TYPE 0  THE EXTENDED EDIT PATTERN FOR FIELD B,COSTS7   WILL BE USED - ZZZ9.9                    .          *****
(R1)         (B,SALES7  )                        (G)                                                                  (ASLGEN'D)    71
** MK4EB09  TYPE 0  THE EXTENDED EDIT PATTERN FOR FIELD B,SALES7   WILL BE USED - ZZZ9.9                    .          *****
(R1)         (B,COSTS8  )                        (G)                                                                  (ASLGEN'D)    72
** MK4EB09  TYPE 0  THE EXTENDED EDIT PATTERN FOR FIELD B,COSTS8   WILL BE USED - ZZZ9.9                    .          *****
(R1)         (B,SALES8  )                        (G)                                                                  (ASLGEN'D)    73
** MK4EB09  TYPE 0  THE EXTENDED EDIT PATTERN FOR FIELD B,SALES8   WILL BE USED - ZZZ9.9                    .          *****
-------------------------------------------------------------------------------------------------------------------------------------
STMT  SEQ                                                                                                      SEQUENCE   LIST
TYPE  NO.                                          TEXT                                                        73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------------
(T1) (101)                (REPORT OF COSTS AND SALES BY DAY WITHIN WEEK - AMOUNTS IN $)                        (ASLGEN'D)    74
(T1) (102)                (1000#                                                       )                       (ASLGEN'D)    75
-------------------------------------------------------------------------------------------------------------------------------------
STMT  SEQ                                                                                                      SEQUENCE   LIST
TYPE  NO.                                          TEXT                                                        73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------------
(F1)                      (COLUMN-HEADING SECTION ##                                   )                       (ASLGEN'D)    76
(F1)                      (#WEEK   ....MONDAY...   ...TUESDAY...   ..WEDNESDAY..#     )                       (ASLGEN'D)    77
(F1)                      (#   ...THURDAY...   ....FRIDAY...   ...SATURDAY..#       )                       (ASLGEN'D)    78
(F1)                      (#   ....SUNDAY...   ..WEEK TOTAL.# ##                     )                       (ASLGEN'D)    79
NOV 06, 2001  12.31.38                                                                                            PAGE      10
-------------------------------------------------------------------------------------------------------------------------------------
STMT  SEQ                                                                                                      SEQUENCE   LIST
TYPE  NO.                                          TEXT                                                        73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------------
(F1)                      (# NO.   COSTS   SALES   COSTS   SALES   COSTS   SALES#     )                       (ASLGEN'D)    80
(F1)                      (#   COSTS   SALES   COSTS   SALES   COSTS   SALES#       )                       (ASLGEN'D)    81
(F1)                      (#   COSTS   SALES   COSTS   SALES# ##                     )                       (ASLGEN'D)    82
```

Figure D-47 Array Examples Application ARRAYAPP: COSTRPT2 Report Procedure Part 6 (Page 2 of 2)

## Array Examples Application ARRAYAPP: ARRIN Procedure – Part 7 of 10

```
NOV 06, 2001  12.31.38                                                                                            PAGE      11
                                                ****************************
                                                *  PROC NAME - ARRIN       *
                                                *  INPUT STREAM PROCEDURE  *
                                                ****************************
-------------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                                MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID          ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM       73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------------
(ER)                                                                                  (S)                           (ASLGEN'D)    83

    ARRIN: PROC TYPE SUBROUTINE
    ;
    ; Read an Array into memory as Cord File 2 - Indexed
    ;
    LASTYEAR: FIELD C 4                  ; ICF Key Field for Cord 2 Read
    ;
    LET T.LASTYEAR = A.YEAR - 1
    ;
    IF T.LASTYEAR = 2.YEAR THEN          ; <- This causes the ICF Read
    ;
    ;  Report Last Year's Array Data
    ;
       REPORT 2.YEAR 2.WEEKNO 2.DAYNO 2.COSTS 2.SALES
              FORMAT WIDTH 80 DATEFMT TODAYX
              GROUP BY 2.YEAR AT LEVEL 1 SUBTITLE NEWPAGE
              ITEM 2.YEAR SPACES 38
       TITLE '                        COST AND SALES BY DAY AND',
             ' WEEK                    '
       TITLE '                       AMOUNTS SHOWN IN 1000 OF D',
```

Figure D-48 Array Examples Application ARRAYAPP: ARRIN Procedure Part 7 (Page 1 of 2)

```
        'OLLARS                '
    TITLE '                ******* LAST YEARS DATA  *',
        '*******                '
    END REPORT
;
ELSE
;
;  There was no data for last year
;
END IF
;
END PROC
;
```

```
STMT   FIELD    FIELD FLD DEC OUTPT EDIT     INITIAL         LINE 1 OF       LINE 2 OF                          SEQUENCE   LIST
TYPE   NAME     LNGTH TYP PLC EDIT  LGTH      VALUE        COLUMN HEADING  COLUMN HEADING                      73 TO 80   SEQ
-------------------------------------------------------------------------------------------------------------------------
(TF) (LASTYEAR) (  4) (C)                                                                                    (ASLGEN'D)    84
-------------------------------------------------------------------------------------------------------------------------
STMT SEQ  LOG CON .....OPERAND-A.....  OPER  .............OPERAND-B.............   .....RESULT.....  PARTIAL FIELD  SEQUENCE   LIST
TYPE NO.  LEV CTR QLF FIELD   SEG-LVL  ATN   QLF FIELD   SEG-LVL  (OR CONSTANT)   QLF FIELD  SEG-LVL  STRT NUMB OPR  73 TO 80   SEQ
-------------------------------------------------------------------------------------------------------------------------
(PR) (100)         (A,YEAR    )       (- )   (D,1                          ) (T,LASTYEAR)                    (ASLGEN'D)    85
(PR) (101)         (T,LASTYEAR)       (EQ)   (2,YEAR    ) 001-1                                            (ASLGEN'D)    86
(PR) (102)                            (NS)     (END                        )                                (ASLGEN'D)    87
(PR) (103)                            (GO)     (SUB REPT__13               )                                (ASLGEN'D)    88
(PR) (104)                            (GO)     (END                        )                                (ASLGEN'D)    89
NOV 06, 2001  12.31.38                                                                                            PAGE    12
                                                  ****************************
                                                  *  REQUEST NAME - REPT__13  *
                                                  *  GENERATED REQUEST        *
                                                  ****************************
-------------------------------------------------------------------------------------------------------------------------
STMT REPORT                                MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID          ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM       73 TO 80   SEQ
-------------------------------------------------------------------------------------------------------------------------
(ER) (TODAYX)                                                                  (S)                               (ASLGEN'D)    90
-------------------------------------------------------------------------------------------------------------------------
 ST SUM V  8  PG  PG   I SP  MAXIMUM TB COLUMN  DT  PG  STRT LIN SL RH .......SUBFILE..... C E A S R S N      C  SEQUENCE   LIST
 TY RPT S LPI WID HGT  M FRM LPP PGS TO HEADING POS POS PAGE NUM UB PN ENT    NAME   BLK/ FMT H F G T S U D      O  73 TO 80   SEQ
 MP     P             G           LT TYP POS         NUM      ML TD REC           BUF     C C S L T M S      N
-------------------------------------------------------------------------------------------------------------------------
(E1)         (080)                                                                                           (ASLGEN'D)    91
-------------------------------------------------------------------------------------------------------------------------
         SP  Q              E N S D C S T C C M M A G D Q PCT/RAT     P               PARTIAL   SEQUENCE   LIST
STMT SEQ BF  L  FIELD   SEG- N P E E T U O U N A I V M E L FIELD   RA- C     PICTURE     STRT NO   73 TO 80   SEQ
TYPE NO. COL F  NAME     LVL D R Q S L B T M T X N G R C F NAME    TIO T      EDIT      CHR CHR
-------------------------------------------------------------------------------------------------------------------------
(R1)      (38) (2,YEAR    ) 001-1          (1) (S)                                                          (ASLGEN'D)    92
(R1)           (2,WEEKNO  ) 002-2                                                                          (ASLGEN'D)    93
(R1)           (2,DAYNO   ) 002-2                                                                          (ASLGEN'D)    94
(R1)           (2,COSTS   ) 002-2                                                                          (ASLGEN'D)    95
(R1)           (2,SALES   ) 002-2                                                                          (ASLGEN'D)    96
-------------------------------------------------------------------------------------------------------------------------
STMT  SEQ                                                                                                    SEQUENCE   LIST
TYPE  NO.                                        TEXT                                                        73 TO 80   SEQ
-------------------------------------------------------------------------------------------------------------------------
(T1)  (101)              (                    COST AND SALES BY DAY AND WEEK    )                         (ASLGEN'D)    97
(T1)  (102)              (                #                                     )                         (ASLGEN'D)    98
(T1)  (103)              (                  AMOUNTS SHOWN IN 1000 OF DOLLARS    )                         (ASLGEN'D)    99
(T1)  (104)              (                #                                     )                         (ASLGEN'D)   100
(T1)  (105)              (                  *******  LAST YEARS DATA  ********  )                         (ASLGEN'D)   101
(T1)  (106)              (                #                                     )                         (ASLGEN'D)   102
```

Figure D-48 Array Examples Application ARRAYAPP: ARRIN Procedure Part 7 (Page 2 of 2)

## Array Examples Application ARRAYAPP: ARROUT Procedure – Part 8 of 10

```
NOV 06, 2001  12.31.38                                                                                            PAGE    13
                                                  ****************************
                                                  *  PROC NAME - ARROUT       *
                                                  *  INPUT STREAM PROCEDURE   *
                                                  ****************************
-------------------------------------------------------------------------------------------------------------------------
STMT REPORT                                MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID          ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM       73 TO 80   SEQ
-------------------------------------------------------------------------------------------------------------------------
(ER)                                                                           (S)                               (ASLGEN'D)   103
```

Figure D-49 Array Examples Application ARRAYAPP: ARROUT Procedure Part 8 (Page 1 of 2)

```
     ARROUT: PROC TYPE SUBROUTINE
     ;
     ; Output the Entire Internal Array to a Subfile
     ; ( This will become Last Year's Data at next Year End )
     ;
      EXTRACT FILE ARRAYOUT ENTIRE A RECFM KEYVSAM
     ;
     END PROC
     ;
-------------------------------------------------------------------------------------------------------------------------------
 ST SUM V  8  PG  PG   I SP  MAXIMUM TB COLUMN  DT  PG  STRT LIN SL RH .......SUBFILE........ C E A S R S N      C  SEQUENCE   LIST
 TY RPT S LPI WID HGT  M FRM LPP PGS TO HEADING POS POS PAGE NUM UB PN ENT    NAME   BLK/ FMT H F G T S U D      O  73 TO 80    SEQ
 MP     P             G            LT TYP POS        NUM      ML TD REC           BUF      C C S L T M S      N
-------------------------------------------------------------------------------------------------------------------------------
(E1)                                                          (NR) (A) (ARRAYOUT)      (K)                      (ASLGEN'D)   104
```

Figure D-49 Array Examples Application ARRAYAPP: ARROUT Procedure Part 8 (Page 2 of 2)

## Array Examples Application ARRAYAPP: HRSERR Procedure – Part 9 of 10

```
NOV 06, 2001  12.31.38                                                                                                  PAGE     14
                                                  ******************************
                                                  *  PROC NAME - HRSERR       *
                                                  *  INPUT STREAM PROCEDURE   *
                                                  ******************************
-------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                                  MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB      SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID            ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM      73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(ER)                                                                                  (S)                            (ASLGEN'D)   105
     HRSERR: REPORT,
     ;
     ; Bad Data Report - List MASTER records with bad Week and Day numbers
     ;
             O.STOREID O.WEEKNO O.DAYNO O.HRSWRKD F.M4OLD F.ASTATUS,
             TYPE SUBROUTINE
             TITLE 'DETAIL LIST OF HOURS WORKED RECORDS WITH'
             TITLE 'BAD DATA IN WEEK NUMBER OR DAY NUMBER FIELDS'
             END REPORT
-------------------------------------------------------------------------------------------------------------------------------
 ST SUM V  8  PG  PG   I SP  MAXIMUM TB COLUMN  DT  PG  STRT LIN SL RH .......SUBFILE........ C E A S R S N      C  SEQUENCE   LIST
 TY RPT S LPI WID HGT  M FRM LPP PGS TO HEADING POS POS PAGE NUM UB PN ENT    NAME   BLK/ FMT H F G T S U D      O  73 TO 80    SEQ
 MP     P             G            LT TYP POS        NUM      ML TD REC           BUF      C C S L T M S      N
-------------------------------------------------------------------------------------------------------------------------------
(E1)                                                                                                                (ASLGEN'D)   106
-------------------------------------------------------------------------------------------------------------------------------
          SP  Q                E  N  S  D  C  S  T  C  C  M  M  A  G  D Q PCT/RAT      P                PARTIAL   SEQUENCE   LIST
STMT SEQ  BF  L  FIELD   SEG-  N  P  E  E  T  U  O  U  N  A  I  V  M  E L  FIELD   RA- C      PICTURE    STRT NO   73 TO 80    SEQ
TYPE NO.  COL F  NAME     LVL  D  R  Q  S  L  B  T  M  T  X  N  G  R  C F  NAME    TIO T       EDIT      CHR CHR
-------------------------------------------------------------------------------------------------------------------------------
(R1)         (O,STOREID ) 001-1                                                                                     (ASLGEN'D)   107
(R1)         (O,WEEKNO  ) 001-1                                                                                     (ASLGEN'D)   108
(R1)         (O,DAYNO   ) 001-1                                                                                     (ASLGEN'D)   109
(R1)         (O,HRSWRKD ) 001-1                                                                                     (ASLGEN'D)   110
(R1)         (F,M4OLD   )                                                                                           (ASLGEN'D)   111
(R1)         (F,ASTATUS )                                                                                           (ASLGEN'D)   112
-------------------------------------------------------------------------------------------------------------------------------
STMT  SEQ                                                                                                             SEQUENCE   LIST
TYPE  NO.                                                  TEXT                                                       73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(T1) (101)                  (DETAIL LIST OF HOURS WORKED RECORDS WITH#                )                          (ASLGEN'D)   113
(T1) (102)                  (BAD DATA IN WEEK NUMBER OR DAY NUMBER FIELDS#            )                          (ASLGEN'D)   114
```

Figure D-50 Array Examples Application ARRAYAPP: HRSERR Procedure Part 9

## Array Examples Application ARRAYAPP: The Reports – Part 10 of 10

```
NOV 06, 2001  12.31.38                                                                                    PAGE     15
** MK4W204  TYPE 0  NUMBER OF MESSAGES PRINTED IS   19.                                                          *****
                   M4OLD    -      364 RECORDS INPUT
                   M4CORD1  -      364 RECORDS INPUT
                   M4CORD2  -        1 RECORDS INPUT
                   M4REPO   -      939 RECORDS OUTPUT
                   M4SUBF1  -        1 RECORDS OUTPUT
                   M4INPUT  -      214 RECORDS INPUT
                   M4LIST   -      604 RECORDS OUTPUT
                     15 TRACKS ASSIGNED TO M4LIB    --   13 TRACKS NOT FULL. LIBRARY DIRECTORY BLOCKING FACTOR IS  800.
** MK4W209  TYPE 0  1008104 BYTES OF MAIN STORAGE UNUSED DURING DECODING PHASE                                  *****
                   1010656 BYTES OF MAIN STORAGE UNUSED DURING COMPILATION PHASE
                    564896 BYTES OF MAIN STORAGE UNUSED DURING PROCESSING PHASE
** MK4ED02  TYPE 0  REQUEST-COSTRPT1  REPORT-1  REQUESTOR ID-                                                  *****
11/06/2001              COST AND SALES BY DAY AND WEEK            PAGE 1
                      AMOUNTS SHOWN IN 1000 OF DOLLARS
-------------------------------------------------------------------------------
  WEEK  DAY   COSTS     SALES
  NO.   NO.   1000'S    1000'S
-------------------------------------------------------------------------------
   01    1       2.2      10.0
   01    2       2.2      10.0
   01    3       2.2      10.0
   01    4       2.2      10.0
   01    5       2.2      10.0
   01    6       2.2      10.0
   01    7       2.2      10.0
   01    8      15.4      70.0
   02    1       2.2       9.8
   02    2       2.2       9.8
   02    3       2.2       9.8
   02    4       2.2       9.8
   02    5       2.2       9.8
   02    6       2.2       9.8
   02    7       2.2       9.8
   02    8      15.4      68.6
   03    1       2.1       9.6
   03    2       2.1       9.6
   03    3       2.1       9.6
   03    4       2.1       9.6
   03    5       2.1       9.6
   03    6       2.1       9.6
   03    7       2.1       9.6
   03    8      14.7      67.2
   04    1       2.1       9.2
   04    2       2.1       9.3
   04    3       2.1       9.3
   04    4       2.1       9.3
   04    5       2.1       9.3
   04    6       2.1       9.3
   04    7       2.1       9.3
   04    8      14.7      65.0
   05    1       2.2       9.7
   05    2       2.2       9.8
   05    3       2.2       9.8
   05    4       2.2       9.8
   05    5       2.2       9.8
   05    6       2.2       9.8
   05    7       2.2       9.8
   05    8      15.4      68.5
   06    1       2.1       9.3
   06    2       2.1       9.3
   06    3       2.1       9.3
   06    4       2.1       9.3
   06    5       2.1       9.3
   06    6       2.1       9.3
   06    7       2.1       9.3
   06    8      14.7      65.1
   07    1       2.1       9.6
   07    2       2.1       9.6
   07    3       2.1       9.6
   07    4       2.1       9.6
   07    5       2.1       9.6
11/06/2001              COST AND SALES BY DAY AND WEEK            PAGE 2
                      AMOUNTS SHOWN IN 1000 OF DOLLARS
-------------------------------------------------------------------------------
  WEEK  DAY   COSTS     SALES
  NO.   NO.   1000'S    1000'S
-------------------------------------------------------------------------------
   07    6       2.1       9.6
   07    7       2.1       9.6
```

Figure D-51 Array Examples Application ARRAYAPP: The Reports Part 10 (Page 1 of 13)

```
  07    8      14.7     67.2
  08    1       2.2      9.8
  08    2       2.2      9.8
  08    3       2.2      9.8
  08    4       2.2      9.8
  08    5       2.2      9.8
  08    6       2.2      9.8
  08    7       2.2      9.8
  08    8      15.4     68.6
  09    1       2.1      9.2
  09    2       2.1      9.2
  09    3       2.1      9.2
  09    4       2.1      9.2
  09    5       2.1      9.2
  09    6       2.1      9.2
  09    7       2.1      9.2
  09    8      14.7     64.4
  10    1       2.1      9.3
  10    2       2.1      9.4
  10    3       2.1      9.4
  10    4       2.1      9.4
  10    5       2.1      9.4
  10    6       2.1      9.4
  10    7       2.1      9.4
  10    8      14.7     65.7
  11    1       2.1      9.4
  11    2       2.1      9.4
  11    3       2.1      9.4
  11    4       2.1      9.4
  11    5       2.1      9.4
  11    6       2.1      9.4
  11    7       2.1      9.5
  11    8      14.7     65.9
  12    1       2.1      9.3
  12    2       2.1      9.3
  12    3       2.1      9.3
  12    4       2.1      9.3
  12    5       2.1      9.3
  12    6       2.1      9.3
  12    7       2.1      9.4
  12    8      14.7     65.2
  13    1       2.2     10.0
  13    2       2.2     10.0
  13    3       2.2     10.0
  13    4       2.2     10.0
  13    5       2.2     10.0
  13    6       2.2     10.0
  13    7       2.2     10.0
  13    8      15.4     70.0
  14    1       2.2      9.9
  14    2       2.2      9.9
11/06/2001            COST AND SALES BY DAY AND WEEK            PAGE 3
                      AMOUNTS SHOWN IN 1000 OF DOLLARS
-------------------------------------------------------------------------------
 WEEK  DAY   COSTS     SALES
 NO.   NO.   1000'S    1000'S
-------------------------------------------------------------------------------
  14    3       2.2      9.9
  14    4       2.2      9.9
  14    5       2.2      9.9
  14    6       2.2      9.9
  14    7       2.2      9.9
  14    8      15.4     69.3
  15    1       2.1      9.5
  15    2       2.2      9.7
  15    3       2.2      9.7
  15    4       2.2      9.7
  15    5       2.2      9.7
  15    6       2.2      9.7
  15    7       2.2      9.7
  15    8      15.3     67.7
  16    1       2.3     10.1
  16    2       2.3     10.1
  16    3       2.3     10.1
  16    4       2.3     10.1
  16    5       2.3     10.1
  16    6       2.3     10.1
  16    7       2.3     10.1
  16    8      16.1     70.7
  17    1       2.1      9.6
  17    2       2.1      9.6
  17    3       2.1      9.6
  17    4       2.1      9.6
  17    5       2.2      9.8
  17    6       2.2      9.8
```

Figure D-51 Array Examples Application ARRAYAPP: The Reports Part 10 (Page 2 of 13)

```
   17    7       2.2       9.8
   17    8      15.0      67.8
   18    1       2.2      10.0
   18    2       2.2      10.0
   18    3       2.2      10.0
   18    4       2.3      10.1
   18    5       2.3      10.1
   18    6       2.3      10.1
   18    7       2.3      10.1
   18    8      15.8      70.4
   19    1       2.3      10.3
   19    2       2.3      10.3
   19    3       2.3      10.3
   19    4       2.3      10.3
   19    5       2.3      10.3
   19    6       2.3      10.3
   19    7       2.3      10.3
   19    8      16.1      72.1
   20    1       2.3      10.5
   20    2       2.4      10.6
   20    3       2.4      10.6
   20    4       2.4      10.6
   20    5       2.4      10.6
   20    6       2.4      10.6
   20    7       2.4      10.6
11/06/2001              COST AND SALES BY DAY AND WEEK            PAGE 4
                        AMOUNTS SHOWN IN 1000 OF DOLLARS
--------------------------------------------------------------------------------
  WEEK  DAY   COSTS     SALES
  NO.   NO.   1000'S    1000'S
--------------------------------------------------------------------------------
   20    8      16.7      74.1
   21    1       2.4      10.6
   21    2       2.4      10.6
   21    3       2.4      10.6
   21    4       2.4      10.6
   21    5       2.4      10.8
   21    6       2.4      10.8
   21    7       2.4      10.8
   21    8      16.8      74.8
   22    1       2.4      10.8
   22    2       2.4      10.8
   22    3       2.4      10.8
   22    4       2.4      10.8
   22    5       2.4      10.8
   22    6       2.4      10.8
   22    7       2.4      10.8
   22    8      16.8      75.6
   23    1       2.4      10.7
   23    2       2.4      10.7
   23    3       2.4      10.7
   23    4       2.4      10.8
   23    5       2.4      10.8
   23    6       2.4      10.8
   23    7       2.4      10.8
   23    8      16.8      75.3
   24    1       2.4      10.7
   24    2       2.4      10.7
   24    3       2.4      10.7
   24    4       2.4      10.7
   24    5       2.4      10.7
   24    6       2.4      10.7
   24    7       2.4      10.7
   24    8      16.8      74.9
   25    1       2.5      11.3
   25    2       2.5      11.3
   25    3       2.5      11.3
   25    4       2.5      11.3
   25    5       2.5      11.3
   25    6       2.5      11.3
   25    7       2.5      11.3
   25    8      17.5      79.1
   26    1       2.5      11.1
   26    2       2.5      11.1
   26    3       2.5      11.1
   26    4       2.5      11.1
   26    5       2.5      11.1
   26    6       2.5      11.1
   26    7       2.5      11.1
   26    8      17.5      77.7
   27    1       2.6      11.5
   27    2       2.6      11.5
   27    3       2.6      11.5
   27    4       2.6      11.5
11/06/2001              COST AND SALES BY DAY AND WEEK            PAGE 5
```

Figure D-51 Array Examples Application ARRAYAPP: The Reports Part 10 (Page 3 of 13)

AMOUNTS SHOWN IN 1000 OF DOLLARS

| WEEK NO. | DAY NO. | COSTS 1000'S | SALES 1000'S |
|---|---|---|---|
| 27 | 5 | 2.6 | 11.5 |
| 27 | 6 | 2.6 | 11.7 |
| 27 | 7 | 2.6 | 11.7 |
| 27 | 8 | 18.2 | 80.9 |
| 28 | 1 | 2.7 | 12.0 |
| 28 | 2 | 2.7 | 12.0 |
| 28 | 3 | 2.7 | 12.0 |
| 28 | 4 | 2.7 | 12.0 |
| 28 | 5 | 2.7 | 12.0 |
| 28 | 6 | 2.7 | 12.1 |
| 28 | 7 | 2.7 | 12.1 |
| 28 | 8 | 18.9 | 84.2 |
| 29 | 1 | 2.6 | 11.6 |
| 29 | 2 | 2.6 | 11.6 |
| 29 | 3 | 2.6 | 11.6 |
| 29 | 4 | 2.6 | 11.6 |
| 29 | 5 | 2.6 | 11.6 |
| 29 | 6 | 2.6 | 11.6 |
| 29 | 7 | 2.6 | 11.6 |
| 29 | 8 | 18.2 | 81.2 |
| 30 | 1 | 2.7 | 11.9 |
| 30 | 2 | 2.7 | 12.0 |
| 30 | 3 | 2.7 | 12.0 |
| 30 | 4 | 2.7 | 12.0 |
| 30 | 5 | 2.7 | 12.0 |
| 30 | 6 | 2.7 | 12.0 |
| 30 | 7 | 2.7 | 12.0 |
| 30 | 8 | 18.9 | 83.9 |
| 31 | 1 | 2.7 | 12.1 |
| 31 | 2 | 2.8 | 12.3 |
| 31 | 3 | 2.8 | 12.3 |
| 31 | 4 | 2.8 | 12.3 |
| 31 | 5 | 2.8 | 12.3 |
| 31 | 6 | 2.8 | 12.3 |
| 31 | 7 | 2.8 | 12.3 |
| 31 | 8 | 19.5 | 85.9 |
| 32 | 1 | 2.8 | 12.4 |
| 32 | 2 | 2.8 | 12.4 |
| 32 | 3 | 2.8 | 12.4 |
| 32 | 4 | 2.8 | 12.4 |
| 32 | 5 | 2.8 | 12.4 |
| 32 | 6 | 2.8 | 12.4 |
| 32 | 7 | 2.8 | 12.4 |
| 32 | 8 | 19.6 | 86.8 |
| 33 | 1 | 3.0 | 13.3 |
| 33 | 2 | 3.0 | 13.3 |
| 33 | 3 | 3.0 | 13.3 |
| 33 | 4 | 3.0 | 13.3 |
| 33 | 5 | 3.0 | 13.3 |
| 33 | 6 | 3.0 | 13.3 |
| 33 | 7 | 3.0 | 13.3 |
| 33 | 8 | 21.0 | 93.1 |
| 34 | 1 | 2.9 | 13.1 |

11/06/2001 COST AND SALES BY DAY AND WEEK PAGE 6

AMOUNTS SHOWN IN 1000 OF DOLLARS

| WEEK NO. | DAY NO. | COSTS 1000'S | SALES 1000'S |
|---|---|---|---|
| 34 | 2 | 3.0 | 13.3 |
| 34 | 3 | 3.0 | 13.3 |
| 34 | 4 | 3.0 | 13.3 |
| 34 | 5 | 3.0 | 13.3 |
| 34 | 6 | 3.0 | 13.3 |
| 34 | 7 | 3.0 | 13.3 |
| 34 | 8 | 20.9 | 92.9 |
| 35 | 1 | 3.0 | 13.2 |
| 35 | 2 | 3.0 | 13.2 |
| 35 | 3 | 3.0 | 13.2 |
| 35 | 4 | 3.0 | 13.2 |
| 35 | 5 | 3.0 | 13.2 |
| 35 | 6 | 3.0 | 13.2 |
| 35 | 7 | 3.0 | 13.2 |
| 35 | 8 | 21.0 | 92.4 |
| 36 | 1 | 3.1 | 13.9 |
| 36 | 2 | 3.1 | 13.9 |
| 36 | 3 | 3.1 | 13.9 |
| 36 | 4 | 3.1 | 13.9 |
| 36 | 5 | 3.1 | 13.9 |
| 36 | 6 | 3.1 | 13.9 |

Figure D-51 Array Examples Application ARRAYAPP: The Reports Part 10 (Page 4 of 13)

```
   36    7       3.1      13.9
   36    8      21.7      97.3
   37    1       3.1      13.8
   37    2       3.1      13.8
   37    3       3.1      13.8
   37    4       3.1      13.8
   37    5       3.1      13.8
   37    6       3.1      13.8
   37    7       3.1      13.8
   37    8      21.7      96.6
   38    1       3.2      14.1
   38    2       3.2      14.1
   38    3       3.2      14.3
   38    4       3.2      14.3
   38    5       3.2      14.3
   38    6       3.2      14.3
   38    7       3.2      14.3
   38    8      22.4      99.7
   39    1       3.3      14.7
   39    2       3.3      14.7
   39    3       3.3      14.7
   39    4       3.3      14.7
   39    5       3.3      14.7
   39    6       3.3      14.7
   39    7       3.3      14.7
   39    8      23.1     102.9
   40    1       3.4      15.0
   40    2       3.4      15.2
   40    3       3.4      15.2
   40    4       3.4      15.2
   40    5       3.4      15.2
   40    6       3.4      15.2
11/06/2001              COST AND SALES BY DAY AND WEEK            PAGE 7
                        AMOUNTS SHOWN IN 1000 OF DOLLARS
-------------------------------------------------------------------------------
  WEEK  DAY   COSTS     SALES
  NO.   NO.   1000'S    1000'S
-------------------------------------------------------------------------------
   40    7       3.4      15.2
   40    8      23.8     106.2
   41    1       3.4      15.4
   41    2       3.5      15.6
   41    3       3.5      15.6
   41    4       3.5      15.6
   41    5       3.5      15.6
   41    6       3.5      15.6
   41    7       3.5      15.6
   41    8      24.4     109.0
   42    1       3.5      15.7
   42    2       3.5      15.7
   42    3       3.5      15.7
   42    4       3.5      15.7
   42    5       3.5      15.7
   42    6       3.5      15.7
   42    7       3.5      15.7
   42    8      24.5     109.9
   43    1       3.5      15.8
   43    2       3.6      16.0
   43    3       3.6      16.0
   43    4       3.6      16.0
   43    5       3.6      16.0
   43    6       3.6      16.0
   43    7       3.6      16.0
   43    8      25.1     111.8
   44    1       3.6      16.2
   44    2       3.6      16.2
   44    3       3.6      16.2
   44    4       3.6      16.2
   44    5       3.6      16.2
   44    6       3.6      16.2
   44    7       3.6      16.2
   44    8      25.2     113.4
   45    1       3.8      16.8
   45    2       3.8      16.8
   45    3       3.8      16.8
   45    4       3.8      16.8
   45    5       3.8      16.8
   45    6       3.8      16.8
   45    7       3.8      16.8
   45    8      26.6     117.6
   46    1       3.8      16.9
   46    2       3.8      16.9
   46    3       3.8      16.9
   46    4       3.8      16.9
   46    5       3.8      16.9
```

Figure D-51 Array Examples Application ARRAYAPP: The Reports Part 10 (Page 5 of 13)

```
  46    6       3.8     16.9
  46    7       3.8     16.9
  46    8      26.6    118.3
  47    1       3.4     15.2
  47    2       3.4     15.2
  47    3       3.4     15.2
11/06/2001              COST AND SALES BY DAY AND WEEK              PAGE 8
                        AMOUNTS SHOWN IN 1000 OF DOLLARS
--------------------------------------------------------------------------------
  WEEK  DAY   COSTS     SALES
  NO.   NO.   1000'S    1000'S
--------------------------------------------------------------------------------
  47    4       3.4     15.2
  47    5       3.4     15.2
  47    6       3.4     15.2
  47    7       3.4     15.2
  47    8      23.8    106.4
  48    1       3.5     15.8
  48    2       3.5     15.8
  48    3       3.5     15.8
  48    4       3.5     15.8
  48    5       3.5     15.8
  48    6       3.5     15.8
  48    7       3.5     15.8
  48    8      24.5    110.6
  49    1       3.5     15.7
  49    2       3.5     15.7
  49    3       3.5     15.7
  49    4       3.5     15.7
  49    5       3.5     15.7
  49    6       3.5     15.7
  49    7       3.5     15.7
  49    8      24.5    109.9
  50    1       3.6     16.1
  50    2       3.6     16.1
  50    3       3.6     16.1
  50    4       3.6     16.1
  50    5       3.6     16.1
  50    6       3.6     16.1
  50    7       3.6     16.1
  50    8      25.2    112.7
  51    1       3.7     16.5
  51    2       3.7     16.5
  51    3       3.7     16.5
  51    4       3.7     16.5
  51    5       3.7     16.5
  51    6       3.7     16.5
  51    7       3.7     16.5
  51    8      25.9    115.5
  52    1       3.8     16.9
  52    2       3.8     16.9
  52    3       3.8     16.9
  52    4       3.8     16.9
  52    5       3.8     16.9
  52    6       3.8     16.9
  52    7       3.8     16.9
  52    8      26.6    118.3
** MK4FT03  TYPE 0  END OF REPORT.                                            *****
** MK4ED02  TYPE 0  REQUEST-COSTRPT2  REPORT-1  REQUESTOR ID-               *****
11/06/01                REPORT OF COSTS AND SALES BY DAY WITHIN WEEK - AMOUNTS IN $1000          PAGE 1
WEEK    ....MONDAY...    ...TUESDAY...   ..WEDNESDAY..   ...THURDAY...    ....FRIDAY...    ...SATURDAY..    ....SUNDAY...    ..WEEK TOTAL.
 NO.   COSTS   SALES   COSTS   SALES   COSTS   SALES   COSTS   SALES   COSTS   SALES   COSTS   SALES   COSTS   SALES   COSTS   SALES
 01      2.2    10.0     2.2    10.0     2.2    10.0     2.2    10.0     2.2    10.0     2.2    10.0     2.2    10.0    15.4    70.0
 02      2.2     9.8     2.2     9.8     2.2     9.8     2.2     9.8     2.2     9.8     2.2     9.8     2.2     9.8    15.4    68.6
 03      2.1     9.6     2.1     9.6     2.1     9.6     2.1     9.6     2.1     9.6     2.1     9.6     2.1     9.6    14.7    67.2
 04      2.1     9.2     2.1     9.3     2.1     9.3     2.1     9.3     2.1     9.3     2.1     9.3     2.1     9.3    14.7    65.0
 05      2.2     9.7     2.2     9.8     2.2     9.8     2.2     9.8     2.2     9.8     2.2     9.8     2.2     9.8    15.4    68.5
 06      2.1     9.3     2.1     9.3     2.1     9.3     2.1     9.3     2.1     9.3     2.1     9.3     2.1     9.3    14.7    65.1
 07      2.1     9.6     2.1     9.6     2.1     9.6     2.1     9.6     2.1     9.6     2.1     9.6     2.1     9.6    14.7    67.2
 08      2.2     9.8     2.2     9.8     2.2     9.8     2.2     9.8     2.2     9.8     2.2     9.8     2.2     9.8    15.4    68.6
 09      2.1     9.2     2.1     9.2     2.1     9.2     2.1     9.2     2.1     9.2     2.1     9.2     2.1     9.2    14.7    64.4
 10      2.1     9.3     2.1     9.4     2.1     9.4     2.1     9.4     2.1     9.4     2.1     9.4     2.1     9.4    14.7    65.7
 11      2.1     9.4     2.1     9.4     2.1     9.4     2.1     9.4     2.1     9.4     2.1     9.4     2.1     9.5    14.7    65.9
 12      2.1     9.3     2.1     9.3     2.1     9.3     2.1     9.3     2.1     9.3     2.1     9.3     2.1     9.4    14.7    65.2
 13      2.2    10.0     2.2    10.0     2.2    10.0     2.2    10.0     2.2    10.0     2.2    10.0     2.2    10.0    15.4    70.0
 14      2.2     9.9     2.2     9.9     2.2     9.9     2.2     9.9     2.2     9.9     2.2     9.9     2.2     9.9    15.4    69.3
 15      2.1     9.5     2.2     9.7     2.2     9.7     2.2     9.7     2.2     9.7     2.2     9.7     2.2     9.7    15.3    67.7
 16      2.3    10.1     2.3    10.1     2.3    10.1     2.3    10.1     2.3    10.1     2.3    10.1     2.3    10.1    16.1    70.7
 17      2.1     9.6     2.1     9.6     2.1     9.6     2.1     9.6     2.2     9.8     2.2     9.8     2.2     9.8    15.0    67.8
 18      2.2    10.0     2.2    10.0     2.2    10.0     2.3    10.1     2.3    10.1     2.3    10.1     2.3    10.1    15.8    70.4
 19      2.3    10.3     2.3    10.3     2.3    10.3     2.3    10.3     2.3    10.3     2.3    10.3     2.3    10.3    16.1    72.1
 20      2.3    10.5     2.4    10.6     2.4    10.6     2.4    10.6     2.4    10.6     2.4    10.6     2.4    10.6    16.7    74.1
 21      2.4    10.6     2.4    10.6     2.4    10.6     2.4    10.6     2.4    10.8     2.4    10.8     2.4    10.8    16.8    74.8
 22      2.4    10.8     2.4    10.8     2.4    10.8     2.4    10.8     2.4    10.8     2.4    10.8     2.4    10.8    16.8    75.6
 23      2.4    10.7     2.4    10.7     2.4    10.7     2.4    10.8     2.4    10.8     2.4    10.8     2.4    10.8    16.8    75.3
```

Figure D-51 Array Examples Application ARRAYAPP: The Reports Part 10 (Page 6 of 13)

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 24 | 2.4 | 10.7 | 2.4 | 10.7 | 2.4 | 10.7 | 2.4 | 10.7 | 2.4 | 10.7 | 2.4 | 10.7 | 2.4 | 10.7 | 16.8 | 74.9 |
| 25 | 2.5 | 11.3 | 2.5 | 11.3 | 2.5 | 11.3 | 2.5 | 11.3 | 2.5 | 11.3 | 2.5 | 11.3 | 2.5 | 11.3 | 17.5 | 79.1 |
| 26 | 2.5 | 11.1 | 2.5 | 11.1 | 2.5 | 11.1 | 2.5 | 11.1 | 2.5 | 11.1 | 2.5 | 11.1 | 2.5 | 11.1 | 17.5 | 77.7 |
| 27 | 2.6 | 11.5 | 2.6 | 11.5 | 2.6 | 11.5 | 2.6 | 11.5 | 2.6 | 11.5 | 2.6 | 11.7 | 2.6 | 11.7 | 18.2 | 80.9 |
| 28 | 2.7 | 12.0 | 2.7 | 12.0 | 2.7 | 12.0 | 2.7 | 12.0 | 2.7 | 12.0 | 2.7 | 12.1 | 2.7 | 12.1 | 18.9 | 84.2 |
| 29 | 2.6 | 11.6 | 2.6 | 11.6 | 2.6 | 11.6 | 2.6 | 11.6 | 2.6 | 11.6 | 2.6 | 11.6 | 2.6 | 11.6 | 18.2 | 81.2 |
| 30 | 2.7 | 11.9 | 2.7 | 12.0 | 2.7 | 12.0 | 2.7 | 12.0 | 2.7 | 12.0 | 2.7 | 12.0 | 2.7 | 12.0 | 18.9 | 83.9 |
| 31 | 2.7 | 12.1 | 2.8 | 12.3 | 2.8 | 12.3 | 2.8 | 12.3 | 2.8 | 12.3 | 2.8 | 12.3 | 2.8 | 12.3 | 19.5 | 85.9 |
| 32 | 2.8 | 12.4 | 2.8 | 12.4 | 2.8 | 12.4 | 2.8 | 12.4 | 2.8 | 12.4 | 2.8 | 12.4 | 2.8 | 12.4 | 19.6 | 86.8 |
| 33 | 3.0 | 13.3 | 3.0 | 13.3 | 3.0 | 13.3 | 3.0 | 13.3 | 3.0 | 13.3 | 3.0 | 13.3 | 3.0 | 13.3 | 21.0 | 93.1 |
| 34 | 2.9 | 13.1 | 3.0 | 13.3 | 3.0 | 13.3 | 3.0 | 13.3 | 3.0 | 13.3 | 3.0 | 13.3 | 3.0 | 13.3 | 20.9 | 92.9 |
| 35 | 3.0 | 13.2 | 3.0 | 13.2 | 3.0 | 13.2 | 3.0 | 13.2 | 3.0 | 13.2 | 3.0 | 13.2 | 3.0 | 13.2 | 21.0 | 92.4 |
| 36 | 3.1 | 13.9 | 3.1 | 13.9 | 3.1 | 13.9 | 3.1 | 13.9 | 3.1 | 13.9 | 3.1 | 13.9 | 3.1 | 13.9 | 21.7 | 97.3 |
| 37 | 3.1 | 13.8 | 3.1 | 13.8 | 3.1 | 13.8 | 3.1 | 13.8 | 3.1 | 13.8 | 3.1 | 13.8 | 3.1 | 13.8 | 21.7 | 96.6 |
| 38 | 3.2 | 14.1 | 3.2 | 14.1 | 3.2 | 14.3 | 3.2 | 14.3 | 3.2 | 14.3 | 3.2 | 14.3 | 3.2 | 14.3 | 22.4 | 99.7 |
| 39 | 3.3 | 14.7 | 3.3 | 14.7 | 3.3 | 14.7 | 3.3 | 14.7 | 3.3 | 14.7 | 3.3 | 14.7 | 3.3 | 14.7 | 23.1 | 102.9 |
| 40 | 3.4 | 15.0 | 3.4 | 15.2 | 3.4 | 15.2 | 3.4 | 15.2 | 3.4 | 15.2 | 3.4 | 15.2 | 3.4 | 15.2 | 23.8 | 106.2 |
| 41 | 3.4 | 15.4 | 3.5 | 15.6 | 3.5 | 15.6 | 3.5 | 15.6 | 3.5 | 15.6 | 3.5 | 15.6 | 3.5 | 15.6 | 24.4 | 109.0 |
| 42 | 3.5 | 15.7 | 3.5 | 15.7 | 3.5 | 15.7 | 3.5 | 15.7 | 3.5 | 15.7 | 3.5 | 15.7 | 3.5 | 15.7 | 24.5 | 109.9 |
| 43 | 3.5 | 15.8 | 3.6 | 16.0 | 3.6 | 16.0 | 3.6 | 16.0 | 3.6 | 16.0 | 3.6 | 16.0 | 3.6 | 16.0 | 25.1 | 111.8 |
| 44 | 3.6 | 16.2 | 3.6 | 16.2 | 3.6 | 16.2 | 3.6 | 16.2 | 3.6 | 16.2 | 3.6 | 16.2 | 3.6 | 16.2 | 25.2 | 113.4 |
| 45 | 3.8 | 16.8 | 3.8 | 16.8 | 3.8 | 16.8 | 3.8 | 16.8 | 3.8 | 16.8 | 3.8 | 16.8 | 3.8 | 16.8 | 26.6 | 117.6 |
| 46 | 3.8 | 16.9 | 3.8 | 16.9 | 3.8 | 16.9 | 3.8 | 16.9 | 3.8 | 16.9 | 3.8 | 16.9 | 3.8 | 16.9 | 26.6 | 118.3 |
| 47 | 3.4 | 15.2 | 3.4 | 15.2 | 3.4 | 15.2 | 3.4 | 15.2 | 3.4 | 15.2 | 3.4 | 15.2 | 3.4 | 15.2 | 23.8 | 106.4 |
| 48 | 3.5 | 15.8 | 3.5 | 15.8 | 3.5 | 15.8 | 3.5 | 15.8 | 3.5 | 15.8 | 3.5 | 15.8 | 3.5 | 15.8 | 24.5 | 110.6 |
| 49 | 3.5 | 15.7 | 3.5 | 15.7 | 3.5 | 15.7 | 3.5 | 15.7 | 3.5 | 15.7 | 3.5 | 15.7 | 3.5 | 15.7 | 24.5 | 109.9 |
| 50 | 3.6 | 16.1 | 3.6 | 16.1 | 3.6 | 16.1 | 3.6 | 16.1 | 3.6 | 16.1 | 3.6 | 16.1 | 3.6 | 16.1 | 25.2 | 112.7 |
| 51 | 3.7 | 16.5 | 3.7 | 16.5 | 3.7 | 16.5 | 3.7 | 16.5 | 3.7 | 16.5 | 3.7 | 16.5 | 3.7 | 16.5 | 25.9 | 115.5 |
| 52 | 3.8 | 16.9 | 3.8 | 16.9 | 3.8 | 16.9 | 3.8 | 16.9 | 3.8 | 16.9 | 3.8 | 16.9 | 3.8 | 16.9 | 26.6 | 118.3 |
| | 142.1 | 634.9 | 142.7 | 636.6 | 142.7 | 636.8 | 142.8 | 637.0 | 142.9 | 637.4 | 142.9 | 637.7 | 142.9 | 637.9 | 999.0 | 4458.3 |

** MK4FT03 TYPE 0 END OF REPORT. *****
** MK4ED02 TYPE 0 REQUEST-REPT__13 REPORT-1 REQUESTOR ID- *****
11/06/2001 PAGE 1

COST AND SALES BY DAY AND WEEK
AMOUNTS SHOWN IN 1000 OF DOLLARS
******* LAST YEARS DATA ********

| WEEK NO. | DAY NO. | COSTS 1000'S | SALES 1000'S |
|---|---|---|---|
| | | 2001 | |
| 01 | 1 | 2.2 | 8.5 |
| 01 | 2 | 2.2 | 8.5 |
| 01 | 3 | 2.2 | 8.5 |
| 01 | 4 | 2.2 | 8.5 |
| 01 | 5 | 2.2 | 8.5 |
| 01 | 6 | 2.2 | 8.5 |
| 01 | 7 | 2.2 | 8.5 |
| 01 | 8 | 15.4 | 59.5 |
| 02 | 1 | 2.1 | 8.3 |
| 02 | 2 | 2.1 | 8.3 |
| 02 | 3 | 2.1 | 8.3 |
| 02 | 4 | 2.1 | 8.3 |
| 02 | 5 | 2.1 | 8.3 |
| 02 | 6 | 2.1 | 8.3 |
| 02 | 7 | 2.1 | 8.3 |
| 02 | 8 | 14.7 | 58.1 |
| 03 | 1 | 2.1 | 8.1 |
| 03 | 2 | 2.1 | 8.1 |
| 03 | 3 | 2.1 | 8.1 |
| 03 | 4 | 2.1 | 8.1 |
| 03 | 5 | 2.1 | 8.1 |
| 03 | 6 | 2.1 | 8.1 |
| 03 | 7 | 2.1 | 8.1 |
| 03 | 8 | 14.7 | 56.7 |
| 04 | 1 | 2.0 | 7.7 |
| 04 | 2 | 2.0 | 7.8 |
| 04 | 3 | 2.0 | 7.8 |
| 04 | 4 | 2.0 | 7.8 |
| 04 | 5 | 2.0 | 7.8 |
| 04 | 6 | 2.0 | 7.8 |
| 04 | 7 | 2.0 | 7.8 |
| 04 | 8 | 14.0 | 54.5 |
| 05 | 1 | 2.1 | 8.2 |
| 05 | 2 | 2.1 | 8.3 |
| 05 | 3 | 2.1 | 8.3 |
| 05 | 4 | 2.1 | 8.3 |
| 05 | 5 | 2.1 | 8.3 |
| 05 | 6 | 2.1 | 8.3 |
| 05 | 7 | 2.1 | 8.3 |
| 05 | 8 | 14.7 | 58.0 |
| 06 | 1 | 2.0 | 7.8 |
| 06 | 2 | 2.0 | 7.8 |
| 06 | 3 | 2.0 | 7.8 |
| 06 | 4 | 2.0 | 7.8 |

Figure D-51 Array Examples Application ARRAYAPP: The Reports Part 10 (Page 7 of 13)

```
  06    5       2.0       7.8
  06    6       2.0       7.8
  06    7       2.0       7.8
11/06/2001                                                          PAGE 2
                      COST AND SALES BY DAY AND WEEK
                     AMOUNTS SHOWN IN 1000 OF DOLLARS
                    *******  LAST YEARS DATA  ********
-----------------------------------------------------------------------------
 WEEK  DAY   COSTS     SALES
 NO.   NO.   1000'S    1000'S
-----------------------------------------------------------------------------
  06    8      14.0      54.6
  07    1       2.1       8.1
  07    2       2.1       8.1
  07    3       2.1       8.1
  07    4       2.1       8.1
  07    5       2.1       8.1
  07    6       2.1       8.1
  07    7       2.1       8.1
  07    8      14.7      56.7
  08    1       2.1       8.3
  08    2       2.1       8.3
  08    3       2.1       8.3
  08    4       2.1       8.3
  08    5       2.1       8.3
  08    6       2.1       8.3
  08    7       2.1       8.3
  08    8      14.7      58.1
  09    1       2.0       7.7
  09    2       2.0       7.7
  09    3       2.0       7.7
  09    4       2.0       7.7
  09    5       2.0       7.7
  09    6       2.0       7.7
  09    7       2.0       7.7
  09    8      14.0      53.9
  10    1       2.0       7.8
  10    2       2.0       7.9
  10    3       2.0       7.9
  10    4       2.0       7.9
  10    5       2.0       7.9
  10    6       2.0       7.9
  10    7       2.0       7.9
  10    8      14.0      55.2
  11    1       2.0       7.9
  11    2       2.0       7.9
  11    3       2.0       7.9
  11    4       2.0       7.9
  11    5       2.0       7.9
  11    6       2.0       7.9
  11    7       2.0       8.0
  11    8      14.0      55.4
  12    1       2.0       7.8
  12    2       2.0       7.8
  12    3       2.0       7.8
  12    4       2.0       7.8
  12    5       2.0       7.8
  12    6       2.0       7.8
  12    7       2.0       7.9
  12    8      14.0      54.7
  13    1       2.2       8.5
11/06/2001                                                          PAGE 3
                      COST AND SALES BY DAY AND WEEK
                     AMOUNTS SHOWN IN 1000 OF DOLLARS
                    *******  LAST YEARS DATA  ********
-----------------------------------------------------------------------------
 WEEK  DAY   COSTS     SALES
 NO.   NO.   1000'S    1000'S
-----------------------------------------------------------------------------
  13    2       2.2       8.5
  13    3       2.2       8.5
  13    4       2.2       8.5
  13    5       2.2       8.5
  13    6       2.2       8.5
  13    7       2.2       8.5
  13    8      15.4      59.5
  14    1       2.1       8.4
  14    2       2.1       8.4
  14    3       2.1       8.4
  14    4       2.1       8.4
  14    5       2.1       8.4
  14    6       2.1       8.4
  14    7       2.1       8.4
  14    8      14.7      58.8
  15    1       2.0       8.0
```

Figure D-51 Array Examples Application ARRAYAPP: The Reports Part 10 (Page 8 of 13)

```
  15    2       2.1       8.2
  15    3       2.1       8.2
  15    4       2.1       8.2
  15    5       2.1       8.2
  15    6       2.1       8.2
  15    7       2.1       8.2
  15    8      14.6      57.2
  16    1       2.2       8.6
  16    2       2.2       8.6
  16    3       2.2       8.6
  16    4       2.2       8.6
  16    5       2.2       8.6
  16    6       2.2       8.6
  16    7       2.2       8.6
  16    8      15.4      60.2
  17    1       2.1       8.1
  17    2       2.1       8.1
  17    3       2.1       8.1
  17    4       2.1       8.1
  17    5       2.1       8.3
  17    6       2.1       8.3
  17    7       2.1       8.3
  17    8      14.7      57.3
  18    1       2.2       8.5
  18    2       2.2       8.5
  18    3       2.2       8.5
  18    4       2.2       8.6
  18    5       2.2       8.6
  18    6       2.2       8.6
  18    7       2.2       8.6
  18    8      15.4      59.9
  19    1       2.2       8.8
  19    2       2.2       8.8
  19    3       2.2       8.8
11/06/2001                                                    PAGE 4
                    COST AND SALES BY DAY AND WEEK
                   AMOUNTS SHOWN IN 1000 OF DOLLARS
                  *******  LAST YEARS DATA  ********
-----------------------------------------------------------------------------
 WEEK  DAY   COSTS     SALES
 NO.   NO.   1000'S    1000'S
-----------------------------------------------------------------------------
  19    4       2.2       8.8
  19    5       2.2       8.8
  19    6       2.2       8.8
  19    7       2.2       8.8
  19    8      15.4      61.6
  20    1       2.3       9.0
  20    2       2.3       9.1
  20    3       2.3       9.1
  20    4       2.3       9.1
  20    5       2.3       9.1
  20    6       2.3       9.1
  20    7       2.3       9.1
  20    8      16.1      63.6
  21    1       2.3       9.1
  21    2       2.3       9.1
  21    3       2.3       9.1
  21    4       2.3       9.1
  21    5       2.3       9.3
  21    6       2.3       9.3
  21    7       2.3       9.3
  21    8      16.1      64.3
  22    1       2.3       9.3
  22    2       2.3       9.3
  22    3       2.3       9.3
  22    4       2.3       9.3
  22    5       2.3       9.3
  22    6       2.3       9.3
  22    7       2.3       9.3
  22    8      16.1      65.1
  23    1       2.3       9.2
  23    2       2.3       9.2
  23    3       2.3       9.2
  23    4       2.3       9.3
  23    5       2.3       9.3
  23    6       2.3       9.3
  23    7       2.3       9.3
  23    8      16.1      64.8
  24    1       2.3       9.2
  24    2       2.3       9.2
  24    3       2.3       9.2
  24    4       2.3       9.2
  24    5       2.3       9.2
  24    6       2.3       9.2
```

Figure D-51 Array Examples Application ARRAYAPP: The Reports Part 10 (Page 9 of 13)

```
  24    7       2.3       9.2
  24    8      16.1      64.4
  25    1       2.4       9.8
  25    2       2.4       9.8
  25    3       2.4       9.8
  25    4       2.4       9.8
  25    5       2.4       9.8
11/06/2001                                                        PAGE 5
                    COST AND SALES BY DAY AND WEEK
                    AMOUNTS SHOWN IN 1000 OF DOLLARS
                   ******* LAST YEARS DATA ********
--------------------------------------------------------------------------------
  WEEK  DAY   COSTS     SALES
  NO.   NO.   1000'S    1000'S
--------------------------------------------------------------------------------
  25    6       2.4       9.8
  25    7       2.4       9.8
  25    8      16.8      68.6
  26    1       2.4       9.6
  26    2       2.4       9.6
  26    3       2.4       9.6
  26    4       2.4       9.6
  26    5       2.4       9.6
  26    6       2.4       9.6
  26    7       2.4       9.6
  26    8      16.8      67.2
  27    1       2.5      10.0
  27    2       2.5      10.0
  27    3       2.5      10.0
  27    4       2.5      10.0
  27    5       2.5      10.0
  27    6       2.5      10.2
  27    7       2.5      10.2
  27    8      17.5      70.4
  28    1       2.6      10.5
  28    2       2.6      10.5
  28    3       2.6      10.5
  28    4       2.6      10.5
  28    5       2.6      10.5
  28    6       2.6      10.6
  28    7       2.6      10.6
  28    8      18.2      73.7
  29    1       2.5      10.1
  29    2       2.5      10.1
  29    3       2.5      10.1
  29    4       2.5      10.1
  29    5       2.5      10.1
  29    6       2.5      10.1
  29    7       2.5      10.1
  29    8      17.5      70.7
  30    1       2.6      10.4
  30    2       2.6      10.5
  30    3       2.6      10.5
  30    4       2.6      10.5
  30    5       2.6      10.5
  30    6       2.6      10.5
  30    7       2.6      10.5
  30    8      18.2      73.4
  31    1       2.6      10.6
  31    2       2.7      10.8
  31    3       2.7      10.8
  31    4       2.7      10.8
  31    5       2.7      10.8
  31    6       2.7      10.8
  31    7       2.7      10.8
11/06/2001                                                        PAGE 6
                    COST AND SALES BY DAY AND WEEK
                    AMOUNTS SHOWN IN 1000 OF DOLLARS
                   ******* LAST YEARS DATA ********
--------------------------------------------------------------------------------
  WEEK  DAY   COSTS     SALES
  NO.   NO.   1000'S    1000'S
--------------------------------------------------------------------------------
  31    8      18.8      75.4
  32    1       2.7      10.9
  32    2       2.7      10.9
  32    3       2.7      10.9
  32    4       2.7      10.9
  32    5       2.7      10.9
  32    6       2.7      10.9
  32    7       2.7      10.9
  32    8      18.9      76.3
  33    1       2.9      11.8
  33    2       2.9      11.8
  33    3       2.9      11.8
```

Figure D-51 Array Examples Application ARRAYAPP: The Reports Part 10 (Page 10 of 13)

```
   33    4       2.9      11.8
   33    5       2.9      11.8
   33    6       2.9      11.8
   33    7       2.9      11.8
   33    8      20.3      82.6
   34    1       2.8      11.6
   34    2       2.9      11.8
   34    3       2.9      11.8
   34    4       2.9      11.8
   34    5       2.9      11.8
   34    6       2.9      11.8
   34    7       2.9      11.8
   34    8      20.2      82.4
   35    1       2.9      11.7
   35    2       2.9      11.7
   35    3       2.9      11.7
   35    4       2.9      11.7
   35    5       2.9      11.7
   35    6       2.9      11.7
   35    7       2.9      11.7
   35    8      20.3      81.9
   36    1       3.0      12.4
   36    2       3.0      12.4
   36    3       3.0      12.4
   36    4       3.0      12.4
   36    5       3.0      12.4
   36    6       3.0      12.4
   36    7       3.0      12.4
   36    8      21.0      86.8
   37    1       3.0      12.3
   37    2       3.0      12.3
   37    3       3.0      12.3
   37    4       3.0      12.3
   37    5       3.0      12.3
   37    6       3.0      12.3
   37    7       3.0      12.3
   37    8      21.0      86.1
   38    1       3.0      12.6
11/06/2001                                                    PAGE 7
                    COST AND SALES BY DAY AND WEEK
                   AMOUNTS SHOWN IN 1000 OF DOLLARS
                  *******  LAST YEARS DATA  ********
------------------------------------------------------------------------
  WEEK  DAY   COSTS     SALES
  NO.   NO.   1000'S    1000'S
------------------------------------------------------------------------
   38    2       3.0      12.6
   38    3       3.1      12.8
   38    4       3.1      12.8
   38    5       3.1      12.8
   38    6       3.1      12.8
   38    7       3.1      12.8
   38    8      21.5      89.2
   39    1       3.2      13.2
   39    2       3.2      13.2
   39    3       3.2      13.2
   39    4       3.2      13.2
   39    5       3.2      13.2
   39    6       3.2      13.2
   39    7       3.2      13.2
   39    8      22.4      92.4
   40    1       3.2      13.5
   40    2       3.3      13.7
   40    3       3.3      13.7
   40    4       3.3      13.7
   40    5       3.3      13.7
   40    6       3.3      13.7
   40    7       3.3      13.7
   40    8      23.0      95.7
   41    1       3.3      13.9
   41    2       3.4      14.1
   41    3       3.4      14.1
   41    4       3.4      14.1
   41    5       3.4      14.1
   41    6       3.4      14.1
   41    7       3.4      14.1
   41    8      23.7      98.5
   42    1       3.4      14.2
   42    2       3.4      14.2
   42    3       3.4      14.2
   42    4       3.4      14.2
   42    5       3.4      14.2
   42    6       3.4      14.2
   42    7       3.4      14.2
   42    8      23.8      99.4
```

Figure D-51 Array Examples Application ARRAYAPP: The Reports Part 10 (Page 11 of 13)

```
  43    1       3.4     14.3
  43    2       3.5     14.5
  43    3       3.5     14.5
  43    4       3.5     14.5
  43    5       3.5     14.5
  43    6       3.5     14.5
  43    7       3.5     14.5
  43    8      24.4    101.3
  44    1       3.5     14.7
  44    2       3.5     14.7
  44    3       3.5     14.7
11/06/2001                                                    PAGE 8
                    COST AND SALES BY DAY AND WEEK
                    AMOUNTS SHOWN IN 1000 OF DOLLARS
                  *******  LAST YEARS DATA  ********
-------------------------------------------------------------------------------
  WEEK  DAY   COSTS     SALES
  NO.   NO.   1000'S    1000'S
-------------------------------------------------------------------------------
  44    4       3.5     14.7
  44    5       3.5     14.7
  44    6       3.5     14.7
  44    7       3.5     14.7
  44    8      24.5    102.9
  45    1       3.6     15.3
  45    2       3.6     15.3
  45    3       3.6     15.3
  45    4       3.6     15.3
  45    5       3.6     15.3
  45    6       3.6     15.3
  45    7       3.6     15.3
  45    8      25.2    107.1
  46    1       3.7     15.4
  46    2       3.7     15.4
  46    3       3.7     15.4
  46    4       3.7     15.4
  46    5       3.7     15.4
  46    6       3.7     15.4
  46    7       3.7     15.4
  46    8      25.9    107.8
  47    1       3.3     13.7
  47    2       3.3     13.7
  47    3       3.3     13.7
  47    4       3.3     13.7
  47    5       3.3     13.7
  47    6       3.3     13.7
  47    7       3.3     13.7
  47    8      23.1     95.9
  48    1       3.4     14.3
  48    2       3.4     14.3
  48    3       3.4     14.3
  48    4       3.4     14.3
  48    5       3.4     14.3
  48    6       3.4     14.3
  48    7       3.4     14.3
  48    8      23.8    100.1
  49    1       3.4     14.2
  49    2       3.4     14.2
  49    3       3.4     14.2
  49    4       3.4     14.2
  49    5       3.4     14.2
  49    6       3.4     14.2
  49    7       3.4     14.2
  49    8      23.8     99.4
  50    1       3.5     14.6
  50    2       3.5     14.6
  50    3       3.5     14.6
  50    4       3.5     14.6
  50    5       3.5     14.6
11/06/2001                                                    PAGE 9
                    COST AND SALES BY DAY AND WEEK
                    AMOUNTS SHOWN IN 1000 OF DOLLARS
                  *******  LAST YEARS DATA  ********
-------------------------------------------------------------------------------
  WEEK  DAY   COSTS     SALES
  NO.   NO.   1000'S    1000'S
-------------------------------------------------------------------------------
  50    6       3.5     14.6
  50    7       3.5     14.6
  50    8      24.5    102.2
  51    1       3.6     15.0
  51    2       3.6     15.0
  51    3       3.6     15.0
  51    4       3.6     15.0
  51    5       3.6     15.0
```

Figure D-51 Array Examples Application ARRAYAPP: The Reports Part 10 (Page 12 of 13)

```
   51    6       3.6      15.0
   51    7       3.6      15.0
   51    8      25.2     105.0
   52    1       3.7      15.4
   52    2       3.7      15.4
   52    3       3.7      15.4
   52    4       3.7      15.4
   52    5       3.7      15.4
   52    6       3.7      15.4
   52    7       3.7      15.4
   52    8      25.9     107.8
** MK4FT03  TYPE 0  END OF REPORT.                                                                                  *****
** MK4ED02  TYPE 0  REQUEST-HRSERR    REPORT-1  REQUESTOR ID-                                                        *****
                                          DETAIL LIST OF HOURS WORKED RECORDS WITH                          PAGE 1
                                        BAD DATA IN WEEK NUMBER OR DAY NUMBER FIELDS
-------------------------------------------------------------------------------------------------------------------------------
  STORE   WEEK  DAY   HOURS        M4OLD       ASTATUS
  ID NO.  NO.   NO.   WORKED
-------------------------------------------------------------------------------------------------------------------------------
      -     -    -         -              -      -
** MK4E203  TYPE 1  NO DATA SELECTED FOR THIS REPORT.                                                                 *****
** MK4FT06  TYPE 1         6 FIELDS WERE PRINTED AS INVALID, MISSING OR OVERFLOW.                                     *****
** MK4FT03  TYPE 0  END OF REPORT.                                                                                    *****
** MK4W204  TYPE 0  NUMBER OF MESSAGES PRINTED IS   10.                                                               *****
                    M4REPO   -      938 RECORDS INPUT
                    M4LIST   -     1028 RECORDS OUTPUT
** MK4W209  TYPE 0   992448 BYTES OF MAIN STORAGE UNUSED DURING REPORTING PHASE                                       *****
```

Figure D-51 Array Examples Application ARRAYAPP: The Reports Part 10 (Page 13 of 13)

## Application To Read An Array As A Master Input File – Banner Page

```
NOV 06, 2001  12.31.43                                                                                    PAGE      1
                                          ********************************************
                                          *  VISION:Builder   4400  (OS/390 - 14.0)  *
                                          *            COPYRIGHT (C) 2002            *
                                          *  COMPUTER ASSOCIATES INTERNATIONAL, INC. *
                                          ********************************************
VISION:Builder JCR1 14.0 110201              ENABLED FOR IBM LANGUAGE ENVIRONMENT                BUILD STAMP = 101306,12:51:24.
                                          COMLIB  MODIFICATIONS (PTFs, APARs) INSTALLED
                                  201,202,203,204,205,206,207,208,209,210,212,213,214,215,216
=================================================================================================================================
=                                                                                                                               =
=                    I N S T A L L A T I O N   P A R A M E T E R S   ( M 4 P A R A M S , M A R K L I B P )                     =
=                                                                                                                               =
=  SYSTEM DELIMITER:   #          PAGE HEIGHT:        66         M4LIST WIDTH:       132        DEF WIDTH OF PAGE:  0        =
=                                                                                                                               =
=  AUTO GRAND:         N          HEADING CHAR:       -          SUBTITLE REPEAT:    N          INVALID FIELD:      *        =
=                                                                                                                               =
=  MISSING FIELD:      -          NON-EDIT FIELD:     +          PERCENT CHAR:       %          LEFT SEPARATOR:     (        =
=                                                                                                                               =
=  RIGHT SEPARATOR:    )          SINGLE SEPARATOR:   ,          SOURCE SPACING:     1          PRINT MESSAGES:     Y        =
=                                                                                                                               =
=  CONSOLE MESSAGES:   N          M4REPO BLOCKSIZE:   4,096      INPUT I/O BUFFERS:  2          OUTPUT I/O BUFFERS: 1        =
=                                                                                                                               =
=  SNGL-STEP STORAGE:  8,192      SNGL-STEP SORTSIZE: 524,288    DIGIT SELECT CHAR:  9          ZERO SUPPRESS CHAR: Z        =
=                                                                                                                               =
=  CURRENCY CHAR:      $          PLUS CHAR:          +          MINUS CHAR:         -          CHECK PROTECT CHAR: *        =
=                                                                                                                               =
=  DECIMAL CHAR:       .          GROUPING CHAR:      ,          PRIMARY PLOT CHAR:  X          SECONDARY PLOT CHAR:*        =
=                                                                                                                               =
=  FIT PLOT CHAR:      .          HORIZONTAL AXIS:    _          HORIZONTAL HASH:    |          VERTICAL AXIS:      |        =
=                                                                                                                               =
=  VERTICAL HASH:      -          MINUTES/HOUR:       60         SECONDS/MINUTE:     60         TIME DELIMITER:     HH:MM:SS =
=                                                                                                                               =
=  DATE FORMAT:      MMM DD, YYYY TODAY FORMAT+DELIM: MM/DD/YY   ISDATE DELIMITER:   YYYY-MM-DD  JULIAN DELIMITER:   YY.DDD   =
=                                                                                                                               =
=  SORT PROGRAM CODE:  2          MINCORE VALUE:      12 K       ALT M4LIST WIDTH:   264        ALT DEF W/OF PAGE:  0        =
=                                                                                                                               =
=  MAX LINES OF TRACE: 1,024      ITEM TRACKING:      0          SUPPRESS NDS REPT?: N          DEFAULT MAXGETMN:   1,024 K  =
=                                                                                                                               =
=  CONDITION CODE 1:   0          CONDITION CODE 2:   4          CONDITION CODE 3:   0          CONDITION CODE 4:   16       =
=                                                                                                                               =
=================================================================================================================================
```

Figure D-52 Application to Read an Array as a Master Input File - Banner Page

## Application To Read An Array As A Master Input File – Part 1 of 2

```
NOV 06, 2001  12.31.43                                                                                                  PAGE      2
-------------------------------------------------------------------------------------------------------------------------------------
  RUN     STMT    FILE    O N S D U T R A B S D S L  RB  R S M R S            START            END              SEQUENCE   LIST
  NAME    TYPE    NAME    L E E I P R E U U R L C S  PL  J O O F S            SEARCH           SEARCH           73 TO 80    SEQ
                          D W Q R D N P D F T M N T  TK  T P P O R
-------------------------------------------------------------------------------------------------------------------------------------
(F366    ) (RC) (FD4ARRAY)(S) (U)      (S)       (T,G)          (R)                                                  (ASLGEN'D)     1

    CONTROL NAME F366 TERM LISTGEN
    FILE MASTER INPUT FD4ARRAY KEYS NONE
    FILE REPORT
    ;
    ; This job processes the array that was produced from
    ; the Array Application and written out as a Subfile.
    ;
    ; We will process Array as if it was a File and Report the Data.
    ; This is done to show you how it works and help visualize
    ; how the Array rows and cells are structured.
    ;
    ;
-------------------------------------------------------------------------------------------------------------------------------------
  RUN     STMT  LOGICAL  LBL  DTF/DD      KEY       KEY       KEY      QUALIFIER   USER I/O  ARRAY      PASSWORD/  SEQUENCE   LIST
  NAME    TYPE  FILENAME       NAME     FIELD-1   FIELD-2   FIELD-3   ICF CRD DBI   MODULE   QUAL       AUTH.-ID   73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------------
(F366    ) (RF) (FD4ARRAY)(S)(M4OLD    )                                                                          (ASLGEN'D)     2
```

Figure D-53 Application to Read an Array as a Master Input File Part 1

## Application To Read An Array As A Master Input File – Part 2 of 2

```
=====================================================================================================================================
=                                                                                                                                   =
=                                                  F I L E   S U M M A R Y                                                          =
=                                                                                                                                   =
= JOBNAME: KREJO04X                              STEPNAME: F336                                                                     =
=                                                                                                                                   =
=  FILE NAME: M4OLD                              DSNAME: BUILDER.R140.ARR2002.R13#1                                                 =
=  FILE NAME: M4LIB                              DSNAME: BUILDER.R140.M4LIB.R13#1                                                   =
=                                                                                                                                   =
=====================================================================================================================================
NOV 06, 2001  12.31.43                                                                                                  PAGE      3
                                                  ******************************
                                                  *  PROC NAME - REPT   1     *
                                                  *  INPUT STREAM PROCEDURE   *
                                                  ******************************
-------------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                                  MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID            ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM       73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------------
(ER)(TODAYX)                                                                          (N)                             (ASLGEN'D)     3
    REPORT O.YEAR O.WEEKNO O.DAYNO O.COSTS O.SALES
      FORMAT WIDTH 80 DATEFMT TODAYX
      GROUP BY O.YEAR
      TITLE 'LIST THE ARRAY THAT WAS OUTPUT'
      TITLE 'AS A SUBFILE - IT IS PROCESSED'
      TITLE 'AS IF IT WERE A TWO LEVEL FILE'
    END REPORT
-------------------------------------------------------------------------------------------------------------------------------------
 ST SUM V  8  PG  PG   I SP  MAXIMUM TB COLUMN  DT  PG  STRT LIN SL RH .......SUBFILE........ C E A S R S N      C  SEQUENCE   LIST
 TY RPT S LPI WID HGT  M FRM LPP PGS TO HEADING POS POS PAGE NUM UB PN ENT    NAME   BLK/ FMT H F G T S U D      O  73 TO 80    SEQ
 MP     P              G            LT TYP POS         NUM      ML TD REC           BUF      C C S L T M S      N
-------------------------------------------------------------------------------------------------------------------------------------
(E1)         (080)                                                                                                  (ASLGEN'D)     4
-------------------------------------------------------------------------------------------------------------------------------------
         SP  Q                E  N  S  D  C  S  T  C  C  M  M  A  G  D Q PCT/RAT      P                PARTIAL   SEQUENCE   LIST
STMT SEQ  BF  L  FIELD   SEG-  N  P  E  E  T  U  O  U  N  A  I  V  M  E L  FIELD   RA- C      PICTURE     STRT NO   73 TO 80    SEQ
TYPE NO.  COL F  NAME      LVL  D  R  Q  S  L  B  T  M  T  X  N  G  R  C F  NAME    TIO T       EDIT       CHR CHR
-------------------------------------------------------------------------------------------------------------------------------------
(R1)         (O,YEAR    ) 001-1           (1)                                                                      (ASLGEN'D)     5
(R1)         (O,WEEKNO  ) 002-2                                                                                      (ASLGEN'D)     6
(R1)         (O,DAYNO   ) 002-2                                                                                      (ASLGEN'D)     7
(R1)         (O,COSTS   ) 002-2                                                                                      (ASLGEN'D)     8
(R1)         (O,SALES   ) 002-2                                                                                      (ASLGEN'D)     9
-------------------------------------------------------------------------------------------------------------------------------------
STMT  SEQ                                                                                                             SEQUENCE   LIST
TYPE  NO.                                                TEXT                                                         73 TO 80    SEQ
```

Figure D-54 Application to Read an Array as a Master Input File Part 2 (Page 1 of 7)

```
-----------------------------------------------------------------------------------------------------------------------------
(T1) (101)                  (LIST THE ARRAY THAT WAS OUTPUT#                          )                        (ASLGEN'D)    10
(T1) (102)                  (AS A SUBFILE - IT IS PROCESSED#                          )                        (ASLGEN'D)    11
(T1) (103)                  (AS IF IT WERE A TWO LEVEL FILE#                          )                        (ASLGEN'D)    12
NOV 06, 2001  12.31.43                                                                                              PAGE      4
** MK4W204  TYPE 0  NUMBER OF MESSAGES PRINTED IS    0.                                                                  *****
                    M4OLD    -        1 RECORDS INPUT
                    M4REPO   -      432 RECORDS OUTPUT
                    M4INPUT  -       19 RECORDS INPUT
                    M4LIST   -      124 RECORDS OUTPUT
                      15 TRACKS ASSIGNED TO M4LIB    --   13 TRACKS NOT FULL. LIBRARY DIRECTORY BLOCKING FACTOR IS  800.
** MK4W209  TYPE 0  1017712 BYTES OF MAIN STORAGE UNUSED DURING DECODING PHASE                                          *****
                    1019288 BYTES OF MAIN STORAGE UNUSED DURING COMPILATION PHASE
                     560120 BYTES OF MAIN STORAGE UNUSED DURING PROCESSING PHASE
** MK4ED02  TYPE 0  REQUEST-REPT__1  REPORT-1  REQUESTOR ID-                                                            *****
11/06/2001               LIST THE ARRAY THAT WAS OUTPUT            PAGE 1
                         AS A SUBFILE - IT IS PROCESSED
                         AS IF IT WERE A TWO LEVEL FILE
-----------------------------------------------------------------------------
  YEAR  WEEK  DAY   COSTS     SALES
  NO.   NO.   NO.   1000'S    1000'S
-----------------------------------------------------------------------------
  2002   01    1       2.2      10.0
         01    2       2.2      10.0
         01    3       2.2      10.0
         01    4       2.2      10.0
         01    5       2.2      10.0
         01    6       2.2      10.0
         01    7       2.2      10.0
         01    8      15.4      70.0
         02    1       2.2       9.8
         02    2       2.2       9.8
         02    3       2.2       9.8
         02    4       2.2       9.8
         02    5       2.2       9.8
         02    6       2.2       9.8
         02    7       2.2       9.8
         02    8      15.4      68.6
         03    1       2.1       9.6
         03    2       2.1       9.6
         03    3       2.1       9.6
         03    4       2.1       9.6
         03    5       2.1       9.6
         03    6       2.1       9.6
         03    7       2.1       9.6
         03    8      14.7      67.2
         04    1       2.1       9.2
         04    2       2.1       9.3
         04    3       2.1       9.3
         04    4       2.1       9.3
         04    5       2.1       9.3
         04    6       2.1       9.3
         04    7       2.1       9.3
         04    8      14.7      65.0
         05    1       2.2       9.7
         05    2       2.2       9.8
         05    3       2.2       9.8
         05    4       2.2       9.8
         05    5       2.2       9.8
         05    6       2.2       9.8
         05    7       2.2       9.8
         05    8      15.4      68.5
         06    1       2.1       9.3
         06    2       2.1       9.3
         06    3       2.1       9.3
         06    4       2.1       9.3
         06    5       2.1       9.3
         06    6       2.1       9.3
         06    7       2.1       9.3
         06    8      14.7      65.1
         07    1       2.1       9.6
         07    2       2.1       9.6
         07    3       2.1       9.6
         07    4       2.1       9.6
11/06/2001               LIST THE ARRAY THAT WAS OUTPUT            PAGE 2
                         AS A SUBFILE - IT IS PROCESSED
                         AS IF IT WERE A TWO LEVEL FILE
-----------------------------------------------------------------------------
  YEAR  WEEK  DAY   COSTS     SALES
  NO.   NO.   NO.   1000'S    1000'S
-----------------------------------------------------------------------------
  2002   07    5       2.1       9.6
         07    6       2.1       9.6
         07    7       2.1       9.6
         07    8      14.7      67.2
```

Figure D-54 Application to Read an Array as a Master Input File Part 2 (Page 2 of 7)

```
        08    1       2.2      9.8
        08    2       2.2      9.8
        08    3       2.2      9.8
        08    4       2.2      9.8
        08    5       2.2      9.8
        08    6       2.2      9.8
        08    7       2.2      9.8
        08    8      15.4     68.6
        09    1       2.1      9.2
        09    2       2.1      9.2
        09    3       2.1      9.2
        09    4       2.1      9.2
        09    5       2.1      9.2
        09    6       2.1      9.2
        09    7       2.1      9.2
        09    8      14.7     64.4
        10    1       2.1      9.3
        10    2       2.1      9.4
        10    3       2.1      9.4
        10    4       2.1      9.4
        10    5       2.1      9.4
        10    6       2.1      9.4
        10    7       2.1      9.4
        10    8      14.7     65.7
        11    1       2.1      9.4
        11    2       2.1      9.4
        11    3       2.1      9.4
        11    4       2.1      9.4
        11    5       2.1      9.4
        11    6       2.1      9.4
        11    7       2.1      9.5
        11    8      14.7     65.9
        12    1       2.1      9.3
        12    2       2.1      9.3
        12    3       2.1      9.3
        12    4       2.1      9.3
        12    5       2.1      9.3
        12    6       2.1      9.3
        12    7       2.1      9.4
        12    8      14.7     65.2
        13    1       2.2     10.0
        13    2       2.2     10.0
        13    3       2.2     10.0
        13    4       2.2     10.0
        13    5       2.2     10.0
        13    6       2.2     10.0
        13    7       2.2     10.0
        13    8      15.4     70.0
11/06/2001              LIST THE ARRAY THAT WAS OUTPUT            PAGE 3
                        AS A SUBFILE - IT IS PROCESSED
                        AS IF IT WERE A TWO LEVEL FILE
--------------------------------------------------------------------------------
  YEAR  WEEK  DAY   COSTS     SALES
  NO.   NO.   NO.   1000'S    1000'S
--------------------------------------------------------------------------------
  2002  14    1       2.2      9.9
        14    2       2.2      9.9
        14    3       2.2      9.9
        14    4       2.2      9.9
        14    5       2.2      9.9
        14    6       2.2      9.9
        14    7       2.2      9.9
        14    8      15.4     69.3
        15    1       2.1      9.5
        15    2       2.2      9.7
        15    3       2.2      9.7
        15    4       2.2      9.7
        15    5       2.2      9.7
        15    6       2.2      9.7
        15    7       2.2      9.7
        15    8      15.3     67.7
        16    1       2.3     10.1
        16    2       2.3     10.1
        16    3       2.3     10.1
        16    4       2.3     10.1
        16    5       2.3     10.1
        16    6       2.3     10.1
        16    7       2.3     10.1
        16    8      16.1     70.7
        17    1       2.1      9.6
        17    2       2.1      9.6
        17    3       2.1      9.6
        17    4       2.1      9.6
        17    5       2.2      9.8
        17    6       2.2      9.8
```

Figure D-54 Application to Read an Array as a Master Input File Part 2 (Page 3 of 7)

```
        17    7       2.2       9.8
        17    8      15.0      67.8
        18    1       2.2      10.0
        18    2       2.2      10.0
        18    3       2.2      10.0
        18    4       2.3      10.1
        18    5       2.3      10.1
        18    6       2.3      10.1
        18    7       2.3      10.1
        18    8      15.8      70.4
        19    1       2.3      10.3
        19    2       2.3      10.3
        19    3       2.3      10.3
        19    4       2.3      10.3
        19    5       2.3      10.3
        19    6       2.3      10.3
        19    7       2.3      10.3
        19    8      16.1      72.1
        20    1       2.3      10.5
        20    2       2.4      10.6
        20    3       2.4      10.6
        20    4       2.4      10.6
11/06/2001              LIST THE ARRAY THAT WAS OUTPUT            PAGE 4
                        AS A SUBFILE - IT IS PROCESSED
                        AS IF IT WERE A TWO LEVEL FILE
--------------------------------------------------------------------------------
  YEAR  WEEK  DAY   COSTS     SALES
  NO.   NO.   NO.   1000'S    1000'S
--------------------------------------------------------------------------------
  2002  20    5       2.4      10.6
        20    6       2.4      10.6
        20    7       2.4      10.6
        20    8      16.7      74.1
        21    1       2.4      10.6
        21    2       2.4      10.6
        21    3       2.4      10.6
        21    4       2.4      10.6
        21    5       2.4      10.8
        21    6       2.4      10.8
        21    7       2.4      10.8
        21    8      16.8      74.8
        22    1       2.4      10.8
        22    2       2.4      10.8
        22    3       2.4      10.8
        22    4       2.4      10.8
        22    5       2.4      10.8
        22    6       2.4      10.8
        22    7       2.4      10.8
        22    8      16.8      75.6
        23    1       2.4      10.7
        23    2       2.4      10.7
        23    3       2.4      10.7
        23    4       2.4      10.8
        23    5       2.4      10.8
        23    6       2.4      10.8
        23    7       2.4      10.8
        23    8      16.8      75.3
        24    1       2.4      10.7
        24    2       2.4      10.7
        24    3       2.4      10.7
        24    4       2.4      10.7
        24    5       2.4      10.7
        24    6       2.4      10.7
        24    7       2.4      10.7
        24    8      16.8      74.9
        25    1       2.5      11.3
        25    2       2.5      11.3
        25    3       2.5      11.3
        25    4       2.5      11.3
        25    5       2.5      11.3
        25    6       2.5      11.3
        25    7       2.5      11.3
        25    8      17.5      79.1
        26    1       2.5      11.1
        26    2       2.5      11.1
        26    3       2.5      11.1
        26    4       2.5      11.1
        26    5       2.5      11.1
        26    6       2.5      11.1
        26    7       2.5      11.1
        26    8      17.5      77.7
11/06/2001              LIST THE ARRAY THAT WAS OUTPUT            PAGE 5
                        AS A SUBFILE - IT IS PROCESSED
                        AS IF IT WERE A TWO LEVEL FILE
--------------------------------------------------------------------------------
```

Figure D-54 Application to Read an Array as a Master Input File Part 2 (Page 4 of 7)

```
 YEAR  WEEK  DAY   COSTS    SALES
 NO.   NO.   NO.   1000'S   1000'S
--------------------------------------------------------------------------------
 2002   27    1       2.6     11.5
        27    2       2.6     11.5
        27    3       2.6     11.5
        27    4       2.6     11.5
        27    5       2.6     11.5
        27    6       2.6     11.7
        27    7       2.6     11.7
        27    8      18.2     80.9
        28    1       2.7     12.0
        28    2       2.7     12.0
        28    3       2.7     12.0
        28    4       2.7     12.0
        28    5       2.7     12.0
        28    6       2.7     12.1
        28    7       2.7     12.1
        28    8      18.9     84.2
        29    1       2.6     11.6
        29    2       2.6     11.6
        29    3       2.6     11.6
        29    4       2.6     11.6
        29    5       2.6     11.6
        29    6       2.6     11.6
        29    7       2.6     11.6
        29    8      18.2     81.2
        30    1       2.7     11.9
        30    2       2.7     12.0
        30    3       2.7     12.0
        30    4       2.7     12.0
        30    5       2.7     12.0
        30    6       2.7     12.0
        30    7       2.7     12.0
        30    8      18.9     83.9
        31    1       2.7     12.1
        31    2       2.8     12.3
        31    3       2.8     12.3
        31    4       2.8     12.3
        31    5       2.8     12.3
        31    6       2.8     12.3
        31    7       2.8     12.3
        31    8      19.5     85.9
        32    1       2.8     12.4
        32    2       2.8     12.4
        32    3       2.8     12.4
        32    4       2.8     12.4
        32    5       2.8     12.4
        32    6       2.8     12.4
        32    7       2.8     12.4
        32    8      19.6     86.8
        33    1       3.0     13.3
        33    2       3.0     13.3
        33    3       3.0     13.3
        33    4       3.0     13.3
11/06/2001              LIST THE ARRAY THAT WAS OUTPUT            PAGE 6
                        AS A SUBFILE - IT IS PROCESSED
                        AS IF IT WERE A TWO LEVEL FILE
--------------------------------------------------------------------------------
 YEAR  WEEK  DAY   COSTS    SALES
 NO.   NO.   NO.   1000'S   1000'S
--------------------------------------------------------------------------------
 2002   33    5       3.0     13.3
        33    6       3.0     13.3
        33    7       3.0     13.3
        33    8      21.0     93.1
        34    1       2.9     13.1
        34    2       3.0     13.3
        34    3       3.0     13.3
        34    4       3.0     13.3
        34    5       3.0     13.3
        34    6       3.0     13.3
        34    7       3.0     13.3
        34    8      20.9     92.9
        35    1       3.0     13.2
        35    2       3.0     13.2
        35    3       3.0     13.2
        35    4       3.0     13.2
        35    5       3.0     13.2
        35    6       3.0     13.2
        35    7       3.0     13.2
        35    8      21.0     92.4
        36    1       3.1     13.9
        36    2       3.1     13.9
        36    3       3.1     13.9
```

Figure D-54 Application to Read an Array as a Master Input File Part 2 (Page 5 of 7)

```
        36    4       3.1     13.9
        36    5       3.1     13.9
        36    6       3.1     13.9
        36    7       3.1     13.9
        36    8      21.7     97.3
        37    1       3.1     13.8
        37    2       3.1     13.8
        37    3       3.1     13.8
        37    4       3.1     13.8
        37    5       3.1     13.8
        37    6       3.1     13.8
        37    7       3.1     13.8
        37    8      21.7     96.6
        38    1       3.2     14.1
        38    2       3.2     14.1
        38    3       3.2     14.3
        38    4       3.2     14.3
        38    5       3.2     14.3
        38    6       3.2     14.3
        38    7       3.2     14.3
        38    8      22.4     99.7
        39    1       3.3     14.7
        39    2       3.3     14.7
        39    3       3.3     14.7
        39    4       3.3     14.7
        39    5       3.3     14.7
        39    6       3.3     14.7
        39    7       3.3     14.7
        39    8      23.1    102.9
11/06/2001              LIST THE ARRAY THAT WAS OUTPUT            PAGE 7
                        AS A SUBFILE - IT IS PROCESSED
                        AS IF IT WERE A TWO LEVEL FILE
-------------------------------------------------------------------------------
  YEAR  WEEK  DAY   COSTS     SALES
  NO.   NO.   NO.   1000'S    1000'S
-------------------------------------------------------------------------------
  2002  40    1       3.4     15.0
        40    2       3.4     15.2
        40    3       3.4     15.2
        40    4       3.4     15.2
        40    5       3.4     15.2
        40    6       3.4     15.2
        40    7       3.4     15.2
        40    8      23.8    106.2
        41    1       3.4     15.4
        41    2       3.5     15.6
        41    3       3.5     15.6
        41    4       3.5     15.6
        41    5       3.5     15.6
        41    6       3.5     15.6
        41    7       3.5     15.6
        41    8      24.4    109.0
        42    1       3.5     15.7
        42    2       3.5     15.7
        42    3       3.5     15.7
        42    4       3.5     15.7
        42    5       3.5     15.7
        42    6       3.5     15.7
        42    7       3.5     15.7
        42    8      24.5    109.9
        43    1       3.5     15.8
        43    2       3.6     16.0
        43    3       3.6     16.0
        43    4       3.6     16.0
        43    5       3.6     16.0
        43    6       3.6     16.0
        43    7       3.6     16.0
        43    8      25.1    111.8
        44    1       3.6     16.2
        44    2       3.6     16.2
        44    3       3.6     16.2
        44    4       3.6     16.2
        44    5       3.6     16.2
        44    6       3.6     16.2
        44    7       3.6     16.2
        44    8      25.2    113.4
        45    1       3.8     16.8
        45    2       3.8     16.8
        45    3       3.8     16.8
        45    4       3.8     16.8
        45    5       3.8     16.8
        45    6       3.8     16.8
        45    7       3.8     16.8
        45    8      26.6    117.6
        46    1       3.8     16.9
```

Figure D-54 Application to Read an Array as a Master Input File Part 2 (Page 6 of 7)

```
        46    2       3.8     16.9
        46    3       3.8     16.9
        46    4       3.8     16.9
11/06/2001             LIST THE ARRAY THAT WAS OUTPUT            PAGE 8
                       AS A SUBFILE - IT IS PROCESSED
                       AS IF IT WERE A TWO LEVEL FILE
--------------------------------------------------------------------------------
  YEAR  WEEK  DAY   COSTS    SALES
  NO.   NO.   NO.   1000'S   1000'S
--------------------------------------------------------------------------------
  2002  46    5       3.8     16.9
        46    6       3.8     16.9
        46    7       3.8     16.9
        46    8      26.6    118.3
        47    1       3.4     15.2
        47    2       3.4     15.2
        47    3       3.4     15.2
        47    4       3.4     15.2
        47    5       3.4     15.2
        47    6       3.4     15.2
        47    7       3.4     15.2
        47    8      23.8    106.4
        48    1       3.5     15.8
        48    2       3.5     15.8
        48    3       3.5     15.8
        48    4       3.5     15.8
        48    5       3.5     15.8
        48    6       3.5     15.8
        48    7       3.5     15.8
        48    8      24.5    110.6
        49    1       3.5     15.7
        49    2       3.5     15.7
        49    3       3.5     15.7
        49    4       3.5     15.7
        49    5       3.5     15.7
        49    6       3.5     15.7
        49    7       3.5     15.7
        49    8      24.5    109.9
        50    1       3.6     16.1
        50    2       3.6     16.1
        50    3       3.6     16.1
        50    4       3.6     16.1
        50    5       3.6     16.1
        50    6       3.6     16.1
        50    7       3.6     16.1
        50    8      25.2    112.7
        51    1       3.7     16.5
        51    2       3.7     16.5
        51    3       3.7     16.5
        51    4       3.7     16.5
        51    5       3.7     16.5
        51    6       3.7     16.5
        51    7       3.7     16.5
        51    8      25.9    115.5
        52    1       3.8     16.9
        52    2       3.8     16.9
        52    3       3.8     16.9
        52    4       3.8     16.9
        52    5       3.8     16.9
        52    6       3.8     16.9
        52    7       3.8     16.9
        52    8      26.6    118.3
** MK4FT03  TYPE 0  END OF REPORT.                                              *****
** MK4W204  TYPE 0  NUMBER OF MESSAGES PRINTED IS    2.                        *****
                    M4REPO   -      431 RECORDS INPUT
                    M4LIST   -      476 RECORDS OUTPUT
** MK4W209  TYPE 0  1005126 BYTES OF MAIN STORAGE UNUSED DURING REPORTING PHASE *****
```

Figure D-54 Application to Read an Array as a Master Input File Part 2 (Page 7 of 7)

# Extended Reporting Capabilities

## Formatted Sectional Report

### Command Group: Summary Section

```
                                          ****************************
                                          *  PROC NAME - TEMP        *
                                          *  INPUT STREAM PROCEDURE  *
                                          ****************************
-------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                              MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID        ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM      73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(ER)                                                                    (N)                          (ASLGEN'D)     8

    TEMP: PROC ; define temp fields                                                                                    9
    COSTVALU: FIELD Z, LEN 7, DEC 2, HEAD 'ESTIMATED' 'COST VALUE'  FLT '$'                                           10
    COSTVDIF: FIELD Z, LEN 9, DEC 2, FLT '$'                                                                          11
    GROUP:    FIELD C, LEN 17                                                                                         12
    NEWCOST:  FIELD Z, LEN 7, DEC 2, HEAD 'EXPECTED' 'NEW COST', FLT '$'                                              13
    NEWPRICE: FIELD Z, LEN 7, DEC 2, HEAD 'RECOMMENDED' 'NEW PRICE', FLT '$'                                          14
    SALEVALU: FIELD Z, LEN 7, DEC 2, HEAD 'ESTIMATED' 'PRICE VALUE' FLT '$'                                           15
    SALEVDIF: FIELD Z, LEN 9, DEC 2, FLT '$'                                                                          16
    END PROC                                                                                                          17
    ;                                                                                                                 18
-------------------------------------------------------------------------------------------------------------------------------
STMT   FIELD    FIELD FLD DEC OUTPT EDIT      INITIAL         LINE 1 OF       LINE 2 OF                            SEQUENCE   LIST
TYPE   NAME     LNGTH TYP PLC EDIT  LGTH       VALUE        COLUMN HEADING  COLUMN HEADING                        73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(TF) (COSTVALU) (  7) (Z) (2) ($  )                      (ESTIMATED     ) (COST VALUE    )                      (ASLGEN'D)    20
(TF) (COSTVDIF) (  9) (Z) (2) ($  )                                                                             (ASLGEN'D)    21
(TF) (GROUP   ) ( 17) (C)                                                                                       (ASLGEN'D)    22
(TF) (NEWCOST ) (  7) (Z) (2) ($  )                      (EXPECTED      ) (NEW COST      )                     (ASLGEN'D)    23
(TF) (NEWPRICE) (  7) (Z) (2) ($  )                      (RECOMMENDED   ) (NEW PRICE     )                     (ASLGEN'D)    24
(TF) (SALEVALU) (  7) (Z) (2) ($  )                      (ESTIMATED     ) (PRICE VALUE   )                     (ASLGEN'D)    25
(TF) (SALEVDIF) (  9) (Z) (2) ($  )                                                                             (ASLGEN'D)    26
** MK4CT06  TYPE 1  REQUEST TEMP     CONTAINS COMMENTS AND TEMPORARY FIELDS ONLY.                                          *****

                                          ****************************
                                          *  PROC NAME - VALUE       *
                                          *  INPUT STREAM PROCEDURE  *
                                          ****************************
-------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                              MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID        ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM      73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(ER)                                                                    (N)                          (ASLGEN'D)    27

    VALUE: PROC                                                                                                       28
    LET T.NEWCOST  = 1.09 * UNITCOST                                                                                  29
    LET T.COSTVALU = T.NEWCOST * QANONHND                                                                             30
    LET T.COSTVDIF = QANONHND * (T.NEWCOST - UNITCOST)                                                                31
    LET T.NEWPRICE = 1.15 * ITMPRICE                                                                                  32
    LET T.SALEVALU = T.NEWPRICE * QANONHND                                                                            33
    LET T.SALEVDIF = QANONHND * (T.NEWPRICE - ITMPRICE)                                                               34
    LET T.GROUP = PF(ITEMNO,1,2) WITH EDIT P'PRODUCT GROUP; ##'                                                       35
     ;                                                                                                                36
     REPORT  T.GROUP, ITEMNO, ITEMNAME,                                                                               37
             T.COSTVALU, T.SALEVALU, T.COSTVDIF, T.SALEVDIF                                                           38
      ITEM T.COSTVDIF, T.SALEVDIF   NOPRINT                                                                           39
         FORMAT DATEFMT DATE                                                                                          40
         TITLE 'COST AND PRICE VALUE'                                                                                 41
         TITLE 'OF STOCK ON HAND'                                                                                     42
         ORDER BY T.GROUP ITEMNO ITEMNAME                                                                             43
         GROUP BY T.GROUP SUBTITLE NEWPAGE                                                                            44
      TOTAL T.COSTVALU, T.SALEVALU, T.COSTVDIF, T.SALEVDIF BY T.GROUP                                                 45
         ;                                                                                                            46
         SECTION SUMMARY                                                                                              47
           SKIP 2 LINES AT LEVEL 1                                                                                    48
           LINE T.GROUP SPACES(4) 'ESTIMATED' SPACES(12) 'COST VALUE'                                                 49
           LINE '      TOTALS' SPACES(11) 'COST VALUE' ,                                                              50
                SPACES(11) 'DIFFERENCE'                                                                               51
           LINE COL(24) '----------' SPACES(11) '----------'                                                          52
           DATA COL(22)   TOTAL(T.COSTVALU) AT LEVEL 1                                                                53
           DATA SPACES(8) TOTAL(T.COSTVDIF) AT LEVEL 1                                                                54
           LINE                    ; output 2 DATA lines                                                              55
           SKIP 1 LINE AT LEVEL 1                                                                                     56
```

Figure D-55 Formatted Sectional Report Command Group: Summary Section (Page 1 of 2)

```
        LINE COL(24) 'ESTIMATED' SPACES(12) 'PRICE VALUE'                                          57
        LINE COL(23) 'PRICE VALUE' SPACES(11) 'DIFFERENCE'                                         58
        LINE COL(23) '-----------' SPACES(11) '----------'                                         59
        DATA COL(22)   TOTAL(T.SALEVALU) AT LEVEL 1                                                60
        DATA SPACES(8) TOTAL(T.SALEVDIF) AT LEVEL 1                                                61
        LINE ;output information from 2 DATA command lines                                         62
      END SECTION                                                                                  63
   END REPORT                                                                                      64
  END PROC                                                                                         65
  ;                                                                                                66

------------------------------------------------------------------------------------------------------------
STMT  FIELD    FIELD FLD DEC OUTPT EDIT     INITIAL       LINE 1 OF      LINE 2 OF              SEQUENCE  LIST
TYPE  NAME     LNGTH TYP PLC EDIT  LGTH      VALUE      COLUMN HEADING  COLUMN HEADING          73 TO 80   SEQ
------------------------------------------------------------------------------------------------------------
(TF) (TEMP__19) (  8) (P) (2)                                                                   (ASLGEN'D)   68
(TF) (TEMP__14) (  8) (P) (2)                                                                   (ASLGEN'D)   69

------------------------------------------------------------------------------------------------------------
STMT SEQ  LOG CON .....OPERAND-A..... OPER ............OPERAND-B............  ......RESULT...... PARTIAL FIELD SEQUENCE  LIST
TYPE NO.  LEV CTR QLF FIELD  SEG-LVL  ATN  QLF FIELD  SEG-LVL  (OR CONSTANT)  QLF FIELD  SEG-LVL STRT NUMB OPR 73 TO 80   SEQ
------------------------------------------------------------------------------------------------------------
(PR)(100)                          (R )  (D,1.09                     ) (T,PTMP__11)             (ASLGEN'D)   70
** MK4DI02  TYPE 0  DEFINITION OF TEMPORARY FIELD PTMP__11 - FORMAT P  LENGTH   8  DECIMALS 2 EDIT LENGTH  21 .       *****
(PR)(101)         (T,PTMP__11)     (* )    (UNITCOST) 001-1          (T,NEWCOST )               (ASLGEN'D)   71
(PR)(102)         (T,NEWCOST )     (* )    (QANONHND) 001-1          (T,COSTVALU)               (ASLGEN'D)   72
(PR)(103)         (T,NEWCOST )     (- )    (UNITCOST) 001-1          (T,TEMP__14)               (ASLGEN'D)   73
(PR)(104)           (QANONHND) 001-1 (* )  (T,TEMP__14)              (T,COSTVDIF)               (ASLGEN'D)   74
(PR)(105)                          (R )  (D,1.15                     ) (T,PTMP__16)             (ASLGEN'D)   75
** MK4DI02  TYPE 0  DEFINITION OF TEMPORARY FIELD PTMP__16 - FORMAT P  LENGTH   8  DECIMALS 2 EDIT LENGTH  21 .       *****
(PR)(106)         (T,PTMP__16)     (* )    (ITMPRICE) 001-1          (T,NEWPRICE)               (ASLGEN'D)   76
(PR)(107)         (T,NEWPRICE)     (* )    (QANONHND) 001-1          (T,SALEVALU)               (ASLGEN'D)   77
(PR)(108)         (T,NEWPRICE)     (- )    (ITMPRICE) 001-1          (T,TEMP__19)               (ASLGEN'D)   78
(PR)(109)           (QANONHND) 001-1 (* )  (T,TEMP__19)              (T,SALEVDIF)               (ASLGEN'D)   79
(PR)(110)           (ITEMNO  ) 001-1 (ED)  (P,PRODUCT GROUP; ##      ) (T,GROUP   )  ( 1) ( 2) (A) (ASLGEN'D)   80
(PR)(111)                          (GO)    (SUB REPT__21             )                          (ASLGEN'D)   81

                                   *****************************
                                   *  REQUEST NAME - REPT__21  *
                                   *  GENERATED REQUEST        *
                                   *****************************

------------------------------------------------------------------------------------------------------------
STMT REPORT                           MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB      SEQUENCE  LIST
TYPE  DATE          REQUESTOR ID      ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM    73 TO 80   SEQ
------------------------------------------------------------------------------------------------------------
(ER)(DATE  )                                                          (S)                       (ASLGEN'D)   82
------------------------------------------------------------------------------------------------------------
 ST SUM V  8  PG  PG   I SP  MAXIMUM TB COLUMN  DT  PG  STRT LIN SL RH .......SUBFILE........ C E A S R S N     C  SEQUENCE  LIST
 TY RPT S LPI WID HGT  M FRM LPP PGS TO HEADING POS POS PAGE NUM UB PN ENT     NAME   BLK/ FMT H F G T S U D     O  73 TO 80   SEQ
 MP     P              G             LT TYP POS         NUM     ML TD REC            BUF      C C S L T M S     N
------------------------------------------------------------------------------------------------------------
(E1)                                                                                            (ASLGEN'D)   83
------------------------------------------------------------------------------------------------------------
          SP  Q                E  N  S  D  C  S  T  C  C  M  M  A  M  D Q PCT/RAT    P              PARTIAL  SEQUENCE  LIST
STMT SEQ  BF  L  FIELD   SEG-  N  P  E  E  T  U  O  U  N  A  I  V  O  E L  FIELD   RA- C    PICTURE  STRT NO  73 TO 80   SEQ
TYPE NO.  COL F  NAME     LVL  D  R  Q  S  L  B  T  M  T  X  N  G  D  C F  NAME    TIO T     EDIT    CHR CHR
------------------------------------------------------------------------------------------------------------
(R1)        (T,GROUP   )         (1)   (1)(P)                                                   (ASLGEN'D)   84
(R1)          (ITEMNO  ) 001-1   (2)                                                            (ASLGEN'D)   85
(R1)          (ITEMNAME) 001-1   (3)                                                            (ASLGEN'D)   86
(R1)        (T,COSTVALU)                (1)                                                     (ASLGEN'D)   87
(R1)        (T,SALEVALU)                (1)                                                     (ASLGEN'D)   88
(R1)        (T,COSTVDIF)       (Y)      (1)                                                     (ASLGEN'D)   89
(R1)        (T,SALEVDIF)       (Y)      (1)                                                     (ASLGEN'D)   90
------------------------------------------------------------------------------------------------------------
STMT  SEQ                                                                                       SEQUENCE  LIST
TYPE  NO.                                  TEXT                                                 73 TO 80   SEQ
------------------------------------------------------------------------------------------------------------
(T1)  (101)               (COST AND PRICE VALUE#                              )                 (ASLGEN'D)   91
(T1)  (102)               (OF STOCK ON HAND#                                  )                 (ASLGEN'D)   92
------------------------------------------------------------------------------------------------------------
STMT  SEQ                                                                                       SEQUENCE  LIST
TYPE  NO.                                  TEXT                                                 73 TO 80   SEQ
------------------------------------------------------------------------------------------------------------
(F1)                      (SUMMARY SECTION ##                                 )                 (ASLGEN'D)   93
(F1)                      (1A 2S##                                            )                 (ASLGEN'D)   94
(F1)                      (T.GROUP# 4B #ESTIMATED# 12B #COST VALUE# ##        )                 (ASLGEN'D)   95
(F1)                      (#     TOTALS# 11B #COST VALUE# 11B #DIFFERENCE# ## )                 (ASLGEN'D)   96
(F1)                      (24C #----------# 11B #----------# ##               )                 (ASLGEN'D)   97
(F1)                      (1L 22C TOTAL T.COSTVALU#1 1L 8B TOTAL T.COSTVDIF#1 ## )              (ASLGEN'D)   98
(F1)                      (1A 1S##                                            )                 (ASLGEN'D)   99
(F1)                      (24C #ESTIMATED# 12B #PRICE VALUE# ##               )                 (ASLGEN'D)  100
(F1)                      (23C #PRICE VALUE# 11B #DIFFERENCE# ##              )                 (ASLGEN'D)  101
(F1)                      (23C #-----------# 11B #----------# ##              )                 (ASLGEN'D)  102
(F1)                      (1L 22C TOTAL T.SALEVALU#1 1L 8B TOTAL T.SALEVDIF#1 ## )              (ASLGEN'D)  103
```

Figure D-55    Formatted Sectional Report Command Group: Summary Section (Page 2 of 2)

## Commands: Summary and Page Title Sections

```
                                        ****************************
                                        *  PROC NAME - RUN         *
                                        *  INPUT STREAM PROCEDURE  *
                                        ****************************
-------------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                              MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID         ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM      73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------------
(ER)                                                                    (N)                          (ASLGEN'D)     9

    RUN: PROC                                                                                                                   10
    SALEAMT: FIELD P, LEN 5, FLT '$'                                                                                            11
      IF ORDERNO EQ ORDERNO  AND  ITEMORD EQ ITEMORD                                                                            12
         LET T.SALEAMT = ITMPRICE * ITMQTYOR                                                                                    13
    ;                                                                                                                           14
      REPORT  ORPERSON CUSTNO CUSTNAME T.SALEAMT  SUMMARYONLY                                                                   15
          FORMAT   DATEFMT DATE   HEADINGS NO                                                                                   16
          ORDER BY ORPERSON CUSTNO                                                                                              17
          GROUP BY ORPERSON        SUBTITLE  NEWPAGE                                                                            18
          GROUP BY CUSTNO CUSTNAME AT LEVEL 2                                                                                   19
          TOTAL T.SALEAMT BY CUSTNO                                                                                             20
        ITEM ORPERSON, CUSTNO, CUSTNAME, T.SALEAMT  NOPRINT                                                                     21
       ;                                                                                                                        22
       SECTION PAGETITLE                                                                                                        23
         LINE COL(5) 'DATE: ' F.DATE SPACES(44) 'PAGE: ' F.PAGE                                                                 24
            SKIP 2 LINES                                                                                                        25
         LINE COL(26) 'SUMMARY OF SALES PER EMPLOYEE'                                                                           26
            SKIP 3 LINES                                                                                                        27
         LINE COL(7) 'SALES PERSON: ' ORPERSON                                                                                  28
            SKIP 4 LINES                                                                                                        29
       END SECTION                                                                                                              30
       ;                                                                                                                        31
       SECTION SUMMARY                                                                                                          32
         DATA COL(4) 'CUSTOMER NUMBER: ' CUSTNO ,                                                                               33
            SPACES(3) 'CUSTOMER NAME: '  CUSTNAME  ,                                                                            34
            SPACES(3) 'TOTAL AMOUNT OF ORDER:  ' ,                                                                              35
                      TOTAL(T.SALEAMT) AT LEVEL 2                                                                               36
         LINE                                                                                                                   37
            SKIP 2 LINES                                                                                                        38
         DATA COL(7) 'TOTAL SALES PER EMPLOYEE:  ' ,                                                                            39
             TOTAL(T.SALEAMT) AT LEVEL 1                                                                                        40
         LINE AT LEVEL 1      ; line always output, but only                                                                    41
       END SECTION            ; only contains data at level 1                                                                   42
      END REPORT                                                                                                                43
    ;                                                                                                                           44
      END IF                                                                                                                    45
    END PROC                                                                                                                    46
    ;                                                                                                                           47
-------------------------------------------------------------------------------------------------------------------------------------
STMT   FIELD    FIELD FLD DEC OUTPT EDIT      INITIAL         LINE 1 OF       LINE 2 OF                          SEQUENCE   LIST
TYPE   NAME     LNGTH TYP PLC EDIT  LGTH       VALUE        COLUMN HEADING  COLUMN HEADING                      73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------------
(TF) (SALEAMT ) (  5) (P)     ($  )                                                                              (ASLGEN'D)    49

-------------------------------------------------------------------------------------------------------------------------------------
STMT SEQ  LOG CON .....OPERAND-A.....  OPER  .............OPERAND-B.............   ......RESULT......  PARTIAL FIELD  SEQUENCE   LIST
TYPE NO.  LEV CTR QLF FIELD   SEG-LVL  ATN   QLF FIELD   SEG-LVL  (OR CONSTANT)   QLF FIELD  SEG-LVL  STRT NUMB OPR 73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------------
(PR) (100)          (ORDERNO ) 002-2  (EQ)     (ORDERNO ) 002-2                                                    (ASLGEN'D)    50
(PR) (101)                            (NS)     (END                          )                                     (ASLGEN'D)    51
(PR) (102)          (ITEMORD ) 005-3  (EQ)     (ITEMORD ) 005-3                                                    (ASLGEN'D)    52
(PR) (103)                            (NS)     (END                          )                                     (ASLGEN'D)    53
(PR) (104)          (ITMPRICE) 005-3  (* )     (ITMQTYOR) 005-3                (T,SALEAMT )                        (ASLGEN'D)    54
(PR) (105)                            (GO)     (SUB REPT___3                 )                                     (ASLGEN'D)    55

                                        ****************************
                                        *  REQUEST NAME - REPT___3 *
                                        *  GENERATED REQUEST  ___  *
                                        ****************************
-------------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                              MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID         ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM      73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------------
(ER) (DATE  )                                                           (S)                          (ASLGEN'D)    56
-------------------------------------------------------------------------------------------------------------------------------------
 ST SUM V  8  PG  PG   I SP  MAXIMUM TB COLUMN  DT  PG  STRT LIN SL RH .......SUBFILE..... C E A S R S N      C  SEQUENCE   LIST
 TY RPT S LPI WID HGT  M FRM LPP PGS TO HEADING POS POS PAGE NUM UB PN ENT    NAME   BLK/ FMT H F G T S U D      O  73 TO 80    SEQ
 MP     P             G            LT TYP POS         NUM      ML TD REC           BUF      C C S L T M S      N
-------------------------------------------------------------------------------------------------------------------------------------
(E1) (Y)                              (X)                                                                       (ASLGEN'D)    57
-------------------------------------------------------------------------------------------------------------------------------------
```

Figure D-56 Formatted Sectional Report Commands: Summary and Page Title Section (Page 1 of 2)

```
         SP  Q                E N S D C S T C C M M A M D Q PCT/RAT      P                PARTIAL   SEQUENCE   LIST
STMT SEQ BF  L  FIELD    SEG-  N P E E T U O U N A I V O E L  FIELD    RA- C     PICTURE    STRT NO   73 TO 80    SEQ
TYPE NO. COL F  NAME      LVL  D R Q S L B T M T X N G D C F  NAME     TIO T      EDIT      CHR CHR
-------------------------------------------------------------------------------------------------------------------------------
(R1)          (ORPERSON) 002-2   (Y) (1)   (1) (P)                                                    (ASLGEN'D)    58
(R1)          (CUSTNO  ) 001-1   (Y) (2)   (2)                                                        (ASLGEN'D)    59
(R1)          (CUSTNAME) 001-1   (Y)       (2)                                                        (ASLGEN'D)    60
(R1)        (T,SALEAMT )         (Y)             (2)                                                  (ASLGEN'D)    61
-------------------------------------------------------------------------------------------------------------------------------
STMT  SEQ
TYPE  NO.                                                TEXT                                          SEQUENCE   LIST
                                                                                                        73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(F1)                      (PAGE-TITLE SECTION ##                                    )                   (ASLGEN'D)    62
(F1)                      (5C #DATE: # F.DATE# 44B #PAGE: # F.PAGE# ##              )                   (ASLGEN'D)    63
(F1)                      (2S##                                                     )                   (ASLGEN'D)    64
(F1)                      (26C #SUMMARY OF SALES PER EMPLOYEE# ##                   )                   (ASLGEN'D)    65
(F1)                      (3S##                                                     )                   (ASLGEN'D)    66
(F1)                      (7C #SALES PERSON: # .ORPERSON# ##                        )                   (ASLGEN'D)    67
(F1)                      (4S##                                                     )                   (ASLGEN'D)    68
(F1)                      (SUMMARY SECTION ##                                       )                   (ASLGEN'D)    69
(F1)                      (2L 4C #CUSTOMER NUMBER: # .CUSTNO# 3B #CUSTOMER NAME: #  )                   (ASLGEN'D)    70
(F1)                      (.CUSTNAME# 3B #TOTAL AMOUNT OF ORDER:  #                 )                   (ASLGEN'D)    71
(F1)                      (TOTAL T.SALEAMT#2 ##                                     )                   (ASLGEN'D)    72
(F1)                      (2S##                                                     )                   (ASLGEN'D)    73
(F1)                      (1A 1L 7C #TOTAL SALES PER EMPLOYEE:  # TOTAL T.SALEAMT#1 )                   (ASLGEN'D)    74
(F1)                      (##                                                       )                   (ASLGEN'D)    75
```

Figure D-56 Formatted Sectional Report Commands: Summary and Page Title Section (Page 2 of 2)

## Commands: Summary Keywords

```
                                                  ****************************
                                                  *  PROC NAME - ORDSTATS    *
                                                  *  INPUT STREAM PROCEDURE  *
                                                  ****************************
-------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                                MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID          ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM       73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(ER)                                                                          (N)                                   (ASLGEN'D)    68

    ORDSTATS: PROC                                                                                                                69
    LASTSPDT: FIELD C LEN 12 INIT 'MAY 15, 2002'                                                                                  70
     REPORT T.LASTSPDT, ITEMNO, ITMPRICE,                                                                                         71
              ORDERNO, ORDQTY, QTYBKORD, QANONHND                                                                                 72
        ITEM T.LASTSPDT NOPRINT                                                                                                   73
        ITEM ITEMNO   SPACES 3                                                                                                    74
        ITEM ORDERNO  SPACES 6                                                                                                    75
        ITEM ORDQTY   SPACES 4                                                                                                    76
        ITEM QTYBKORD SPACES 11                                                                                                   77
        ITEM QANONHND SPACES 10                                                                                                   78
        FORMAT DATEFMT DATE                                                                                                       79
        ORDER BY ITEMNO                                                                                                           80
        TOTAL   ORDQTY   AT LEVEL G                                                                                               81
        TOTAL   QTYBKORD AT LEVEL G                                                                                               82
        AVERAGE QTYBKORD AT LEVEL G                                                                                               83
        MAX     ORDQTY   AT LEVEL G                                                                                               84
        MIN     ORDQTY   AT LEVEL G                                                                                               85
        RATIO   QANONHND TO ORDQTY AT LEVEL G                                                                                     86
        PERCENT QTYBKORD OF ORDQTY AT LEVEL G                                                                                     87
     ;                                                                                                                            88
       SECTION PAGETITLE                                                                                                          89
         SKIP 2 LINES                                                                                                             90
         LINE 'DATE:  ' F.DATE  SPACES(40) 'PAGE:  ' F.PAGE                                                                       91
         SKIP 1 LINE                                                                                                              92
         LINE COL(20) 'INVENTORY AND ORDERING STATISTICS'                                                                         93
       END SECTION                                                                                                                94
     ;                                                                                                                            95
       SECTION COLUMNHEADING                                                                                                      96
         LINE COL(5) 'ITEM       ITEM       ORDER    QUANTITY',                                                                   97
              '      QUANTITY      QUANTITY'                                                                                      98
         LINE COL(4) 'NUMBER      PRICE      NUMBER   ORDERED' ,                                                                  99
              '     BACK ORDERED     ON HAND'                                                                                    100
         LINE COL(4) '-------   --------     ------   --------',                                                                 101
              '    ------------    --------'                                                                                     102
       END SECTION                                                                                                               103
     ;                                                                                                                           104
       SECTION SUMMARY                                                                                                           105
         SKIP 5 LINES                                                                                                            106
```

Figure D-57 Formatted Sectional Report Commands: Summary Keywords (Page 1 of 2)

```
        LINE 'INVENTORY STATS YTD TO:  ' T.LASTSPDT    AT LEVEL G                                    107
        LINE 'TOTAL QUANTITY ORDERED: ' TOTAL(ORDQTY)  ,                                             108
      '     TOTAL BACK ORDERED: ' TOTAL(QTYBKORD),                                                  109
                                                       AT LEVEL G                                    110
        LINE 'MAX ORDERED - ANY ITEM:' MAX(ORDQTY),                                                  111
      '      MIN ORDERED - ANY ITEM:' MIN(ORDQTY),                                                   112
                                                       AT LEVEL G                                    113
        LINE 'RATIO OF TOTAL QUANTITY ON HAND TO ORDERS:'  ,                                         114
                                  RATIO(QANONHND)      AT LEVEL G                                    115
        LINE 'PERCENTAGE OF TOTAL BACK ORDERED TO ',                                                 116
                   'TOTAL ORDERS:' PERCENT(QTYBKORD) AT LEVEL G                                      117
        LINE 'AVERGE QUANTITY BACK ORDERED:' AVERAGE(QTYBKORD),                                      118
                                                       AT LEVEL G                                    119
      END SECTION                                                                                    120
    END REPORT                                                                                       121
  END PROC                                                                                           122
-------------------------------------------------------------------------------------------------------
STMT   FIELD    FIELD FLD DEC OUTPT EDIT      INITIAL          LINE 1 OF       LINE 2 OF                    SEQUENCE   LIST
TYPE   NAME     LNGTH TYP PLC EDIT  LGTH       VALUE         COLUMN HEADING  COLUMN HEADING                73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------
(TF) (LASTSPDT) ( 12) (C)               (MAY 15, 2002    )                                    (ASLGEN'D)   123
-------------------------------------------------------------------------------------------------------
STMT SEQ  LOG CON .....OPERAND-A.....  OPER  .............OPERAND-B.............   .....RESULT......  PARTIAL FIELD  SEQUENCE   LIST
TYPE NO.  LEV CTR QLF FIELD   SEG-LVL  ATN   QLF FIELD   SEG-LVL  (OR CONSTANT)   QLF FIELD  SEG-LVL  STRT NUMB OPR  73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------
(PR) (100)                             (GO)      (SUB REPT__11                 )                    (ASLGEN'D)   124

                                                  ****************************
                                                  *  REQUEST NAME - REPT__11  *
                                                  *  GENERATED REQUEST        *
                                                  ****************************
-------------------------------------------------------------------------------------------------------
STMT REPORT                              MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID         ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM       73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------
(ER) (DATE  )                                                                  (S)                   (ASLGEN'D)   125
-------------------------------------------------------------------------------------------------------
 ST SUM V  8  PG  PG   I SP  MAXIMUM TB COLUMN  DT  PG  STRT LIN SL RH .......SUBFILE..... C E A S R S N      C  SEQUENCE   LIST
 TY RPT S LPI WID HGT  M FRM LPP PGS TO HEADING POS POS PAGE NUM UB PN ENT    NAME   BLK/ FMT H F G T S U D      O  73 TO 80    SEQ
 MP     P             G            LT TYP POS         NUM      ML TD REC           BUF      C C S L T M S      N
-------------------------------------------------------------------------------------------------------
(E1)                                                                                                 (ASLGEN'D)   126
-------------------------------------------------------------------------------------------------------
         SP  Q                E N S  D  C  S  T  C  C  M  M  A  M  D Q PCT/RAT      P                PARTIAL   SEQUENCE   LIST
STMT SEQ BF  L  FIELD   SEG-  N P E  E  T  U  O  U  N  A  I  V  O  E L  FIELD   RA- C     PICTURE     STRT NO   73 TO 80    SEQ
TYPE NO. COL F  NAME      LVL D R Q  S  L  B  T  M  T  X  N  G  D  C F  NAME    TIO T      EDIT       CHR CHR
-------------------------------------------------------------------------------------------------------
(R1)         (T,LASTSPDT)         (Y)                                                                (ASLGEN'D)   127
(R1)     ( 3)  (ITEMNO  ) 001-1      (1)                                                            (ASLGEN'D)   128
(R1)           (ITMPRICE) 001-1                                                                     (ASLGEN'D)   129
(R1)     ( 6)  (ORDERNO ) 002-2                                                                     (ASLGEN'D)   130
(R1)     ( 4)  (ORDQTY  ) 001-1                 (G)       (G) (G)                                   (ASLGEN'D)   131
(R1)     (11)  (QTYBKORD) 002-2                 (G)             (G)       (ORDQTY  )   (G)          (ASLGEN'D)   132
(R1)     (10)  (QANONHND) 001-1                                           (ORDQTY  ) (G)            (ASLGEN'D)   133
-------------------------------------------------------------------------------------------------------
STMT  SEQ                                                                                            SEQUENCE   LIST
TYPE  NO.                                          TEXT                                              73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------
(F1)                      (PAGE-TITLE SECTION ##                                    )                (ASLGEN'D)   134
(F1)                      (2S##                                                     )                (ASLGEN'D)   135
(F1)                      (#DATE:  # F.DATE# 40B #PAGE:  # F.PAGE# ##               )                (ASLGEN'D)   136
(F1)                      (1S##                                                     )                (ASLGEN'D)   137
(F1)                      (20C #INVENTORY AND ORDERING STATISTICS# ##               )                (ASLGEN'D)   138
(F1)                      (COLUMN-HEADING SECTION ##                                )                (ASLGEN'D)   139
(F1)                      (5C #ITEM       ITEM       ORDER    QUANTITY#            )                (ASLGEN'D)   140
(F1)                      (#      QUANTITY      QUANTITY# ##                        )                (ASLGEN'D)   141
(F1)                      (4C #NUMBER      PRICE      NUMBER   ORDERED#           )                (ASLGEN'D)   142
(F1)                      (#     BACK ORDERED     ON HAND# ##                      )                (ASLGEN'D)   143
(F1)                      (4C #-------   --------     ------   --------#           )                (ASLGEN'D)   144
(F1)                      (#     ------------    --------# ##                      )                (ASLGEN'D)   145
(F1)                      (SUMMARY SECTION ##                                       )                (ASLGEN'D)   146
(F1)                      (5S##                                                     )                (ASLGEN'D)   147
(F1)                      (GA #INVENTORY STATS YTD TO:  # T.LASTSPDT# ##            )                (ASLGEN'D)   148
(F1)                      (GA #TOTAL QUANTITY ORDERED: # TOTAL .ORDQTY  #G          )                (ASLGEN'D)   149
(F1)                      (#     TOTAL BACK ORDERED: # TOTAL .QTYBKORD#G ##        )                (ASLGEN'D)   150
(F1)                      (GA #MAX ORDERED - ANY ITEM:# MAX .ORDQTY  #G             )                (ASLGEN'D)   151
(F1)                      (#      MIN ORDERED - ANY ITEM:# MIN .ORDQTY  #G ##       )                (ASLGEN'D)   152
(F1)                      (GA #RATIO OF TOTAL QUANTITY ON HAND TO ORDERS:#          )                (ASLGEN'D)   153
(F1)                      (RATIO .QANONHND#G ##                                     )                (ASLGEN'D)   154
(F1)                      (GA #PERCENTAGE OF TOTAL BACK ORDERED TO # #TOTAL ORDERS:# )               (ASLGEN'D)   155
(F1)                      (PCT .QTYBKORD#G ##                                       )                (ASLGEN'D)   156
(F1)                      (GA #AVERGE QUANTITY BACK ORDERED:# AVG .QTYBKORD#G ##     )               (ASLGEN'D)   157
```

Figure D-57 Formatted Sectional Report Commands: Summary Keywords (Page 2 of 2)

## Commands: Computed Summaries

```
                                                *****************************
                                                *  PROC NAME - COMPUTE      *
                                                *  INPUT STREAM PROCEDURE   *
                                                *****************************
-----------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                                   MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ         BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID             ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM      73 TO 80    SEQ
-----------------------------------------------------------------------------------------------------------------------------------
(ER)                                                                                (N)                              (ASLGEN'D)    76

     COMPUTE: PROC                                                                                                                 77
        IF ORDERNO = ORDERNO    ; only report customers with orders                                                                78
              LET T.TOTITMPC = QTYSHIP * ITMPRICE                                                                                  79
              LET T.FREIGHT  = FRTCOST + T.FREIGHT                                                                                 80
     ;                                                                                                                             81
     REPORT CUSTNAME CUSTNO CUSTPH INVNO T.FREIGHT  ,                                                                              82
            ORDERNO ITEMORD ITMQTYOR ITMPRICE T.TOTITMPC                                                                           83
       FORMAT  DATEFMT DATE                                                                                                        84
       ORDER BY CUSTNAME, INVNO, ORDERNO                                                                                           85
       GROUP BY CUSTNAME        SUBTITLE  NEWPAGE                                                                                  86
       GROUP BY CUSTNO, CUSTPH, INVNO, ORDERNO  AT LEVEL 2                                                                         87
       TOTAL T.FREIGHT  T.TOTITMPC  BY CUSTNAME                                                                                    88
       ITEM CUSTNAME, CUSTNO, INVNO  NOPRINT                                                                                       89
       ITEM CUSTPH     NOPRINT  PICTURE P'(###) ###-####'                                                                          90
       ITEM T.FREIGHT  NOPRINT  PICTURE P'$ZZZZZ9.99'                                                                              91
       ITEM ORDERNO  SPACES 3                                                                                                      92
       ITEM ITMQTYOR SPACES 0                                                                                                      93
       ITEM ITEMORD  SPACES 5                                                                                                      94
       ITEM ITMPRICE SPACES 1    PICTURE P'$$$$$9.99'                                                                              95
       ITEM T.TOTITMPC SPACES 4  PICTURE P'$ZZZZZ9.99'                                                                             96
      ;                                                                                                                            97
      SECTION PAGETITLE                                                                                                            98
              SKIP 2 LINES                                                                                                         99
          LINE 'INVOICE NO: ' INVNO   COL(50) F.DATE                                                                              100
              SKIP 1 LINES                                                                                                        101
          LINE COL(8) CUSTNAME                                                                                                    102
          LINE COL(8) 'CUSTOMER ID:  ' CUSTNO                                                                                     103
          LINE COL(8) CUSTPH                                                                                                      104
              SKIP 2 LINES                                                                                                        105
          LINE COL(63) 'PAGE: ' F.PAGE                                                                                            106
              SKIP 2 LINES                                                                                                        107
      END SECTION                                                                                                                 108
      ;                                                                                                                           109
      SECTION COLUMNHEADING                                                                                                       110
          LINE LIT('*',70)                                                                                                        111
          LINE '** ORDER ' SPACES(4) ' ITEM ' SPACES(4) 'QUANTITY' ,                                                              112
               SPACES(4) 'ITEM ' SPACES(8) 'TOTAL' COL(69) '**'                                                                   113
          LINE '** NUMBER' SPACES(4) 'NUMBER' SPACES(4) ' ORDERED' ,                                                              114
               SPACES(4) 'PRICE' SPACES(8) ' COST'  COL(69) '**'                                                                  115
          LINE LIT('*',70)                                                                                                        116
      END SECTION                                                                                                                 117
      ;                                                                                                                           118
      SECTION SUMMARY                                                                                                             119
        COMPUTE STEMP01 = TOTAL(T.TOTITMPC) * .0725  ,                                                                            120
                ROUNDED, DEC 2, AT LEVEL 1, PIC P'$ZZZZZ9.99'                                                                     121
        COMPUTE STEMP02 = TOTAL(T.TOTITMPC) + STEMP01,                                                                            122
                ROUNDED, DEC 2, AT LEVEL 1                                                                                        123
        COMPUTE STEMP03 = TOTAL(T.FREIGHT)  + STEMP02,                                                                            124
                ROUNDED, DEC 2, AT LEVEL 1, PIC P'$ZZZZZ9.99'                                                                     125
        SKIP 3 LINES AT LEVEL 1                                                                                                   126
        LINE COL(40) 'SUBTOTAL     ' TOTAL(T.TOTITMPC) AT LEVEL 1                                                                 127
        LINE COL(40) 'SALES TAX      ' STEMP01          AT LEVEL 1                                                                128
        LINE COL(40) 'FREIGHT CHARGE ' TOTAL(T.FREIGHT)  AT LEVEL 1                                                               129
        LINE COL(56) LIT('-',9)                         AT LEVEL 1                                                                130
        LINE COL(40) 'INVOICE TOTAL  ' STEMP03          AT LEVEL 1                                                                131
        LINE COL(56) LIT('=',9)                         AT LEVEL 1                                                                132
      END SECTION                                                                                                                 133
     END REPORT                                                                                                                   134
     END IF                                                                                                                       135
     END PROC                                                                                                                     136
-----------------------------------------------------------------------------------------------------------------------------------
STMT SEQ  LOG CON .....OPERAND-A.....  OPER  ............OPERAND-B............   ......RESULT......  PARTIAL FIELD  SEQUENCE   LIST
TYPE NO.  LEV CTR QLF FIELD   SEG-LVL  ATN   QLF FIELD   SEG-LVL  (OR CONSTANT)   QLF FIELD  SEG-LVL  STRT NUMB OPR  73 TO 80    SEQ
-----------------------------------------------------------------------------------------------------------------------------------
(PR)(100)          (ORDERNO ) 002-2  (EQ)    (ORDERNO ) 002-2                                                   (ASLGEN'D)   137
(PR)(101)                            (NS)    (END                                )                              (ASLGEN'D)   138
(PR)(102)          (QTYSHIP ) 004-4  (* )    (ITMPRICE) 005-3                      (T,TOTITMPC)                 (ASLGEN'D)   139
** MK4DI02  TYPE 0  DEFINITION OF TEMPORARY FIELD TOTITMPC - FORMAT Z  LENGTH   7  DECIMALS 0 EDIT LENGTH  10 .              *****
(PR)(103)          (FRTCOST ) 003-3  (+ )   (T,FREIGHT )                           (T,FREIGHT )                 (ASLGEN'D)   140
** MK4DI02  TYPE 0  DEFINITION OF TEMPORARY FIELD FREIGHT  - FORMAT P  LENGTH   4  DECIMALS 2 EDIT LENGTH  10 .              *****
(PR)(104)                            (GO)    (SUB REPT__11                       )                              (ASLGEN'D)   141
```

Figure D-58 Formatted Sectional Report Commands: Computed Summaries (Page 1 of 2)

```
                                    ****************************
                                    *  REQUEST NAME - REPT_11  *
                                    *  GENERATED REQUEST  ___  *
                                    ****************************

-------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                                MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID           ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM      73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(ER) (DATE  )                                                                    (S)                                (ASLGEN'D)   142
-------------------------------------------------------------------------------------------------------------------------------
 ST SUM V  8  PG  PG   I SP  MAXIMUM TB COLUMN  DT  PG  STRT LIN SL RH .......SUBFILE..... C E A S R S N      C  SEQUENCE   LIST
 TY RPT S LPI WID HGT  M FRM LPP PGS TO HEADING POS POS PAGE NUM UB PN ENT    NAME   BLK/ FMT H F G T S U D      O  73 TO 80    SEQ
 MP     P             G            LT TYP POS         NUM      ML TD REC           BUF      C C S L T M S      N
-------------------------------------------------------------------------------------------------------------------------------
(E1)                                                                                                            (ASLGEN'D)   143
-------------------------------------------------------------------------------------------------------------------------------
         SP  Q                E  N  S  D  C  S  T  C  C  M  M  A  M  D Q PCT/RAT      P                 PARTIAL   SEQUENCE   LIST
STMT SEQ BF  L  FIELD   SEG-  N  P  E  E  T  U  O  U  N  A  I  V  O  E L  FIELD   RA- C      PICTURE     STRT NO  73 TO 80    SEQ
TYPE NO. COL F  NAME      LVL D  R  Q  S  L  B  T  M  T  X  N  G  D  C F  NAME    TIO T       EDIT       CHR CHR
-------------------------------------------------------------------------------------------------------------------------------
(R1)           (CUSTNAME) 001-1   (Y) (1)   (1) (P)                                                                (ASLGEN'D)   144
(R1)           (CUSTNO  ) 001-1   (Y)       (2)                                                                    (ASLGEN'D)   145
(R1)           (CUSTPH  ) 001-1   (Y)       (2)                                                                    (ASLGEN'D)   146
** MK4FW09  TYPE 0  THE EXTENDED EDIT PATTERN FOR FIELD  ,CUSTPH   WILL BE USED - (###) ###-####              .                 *****
(R1)           (INVNO   ) 003-3   (Y) (2)   (2)                                                                    (ASLGEN'D)   147
(R1)         (T,FREIGHT )         (Y)           (1)                                                               (ASLGEN'D)   148
** MK4FW09  TYPE 0  THE EXTENDED EDIT PATTERN FOR FIELD T,FREIGHT  WILL BE USED - $ZZZZZ9.99                  .                 *****
(R1)      ( 3)  (ORDERNO ) 002-2       (3)   (2)                                                                    (ASLGEN'D)   149
(R1)      ( 5)  (ITEMORD ) 005-3                                                                                    (ASLGEN'D)   150
(R1)      ( 0)  (ITMQTYOR) 005-3                                                                                    (ASLGEN'D)   151
(R1)      ( 1)  (ITMPRICE) 005-3                                                                                    (ASLGEN'D)   152
** MK4FW09  TYPE 0  THE EXTENDED EDIT PATTERN FOR FIELD  ,ITMPRICE WILL BE USED - $$$$$9.99                .                 *****
(R1)      ( 4) (T,TOTITMPC)                    (1)                                                               (ASLGEN'D)   153
** MK4FW09  TYPE 0  THE EXTENDED EDIT PATTERN FOR FIELD T,TOTITMPC WILL BE USED - $ZZZZZ9.99                 .                 *****
-------------------------------------------------------------------------------------------------------------------------------
STMT  SEQ                                                                                                        SEQUENCE   LIST
TYPE  NO.                                          TEXT                                                          73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(F1)                      (PAGE-TITLE SECTION ##                                    )                            (ASLGEN'D)   154
(F1)                      (2S##                                                     )                            (ASLGEN'D)   155
(F1)                      (#INVOICE NO: # .INVNO# 50C F.DATE# ##                    )                            (ASLGEN'D)   156
(F1)                      (1S##                                                     )                            (ASLGEN'D)   157
(F1)                      (8C .CUSTNAME# ##                                         )                            (ASLGEN'D)   158
(F1)                      (8C #CUSTOMER ID:  # .CUSTNO# ##                          )                            (ASLGEN'D)   159
(F1)                      (8C .CUSTPH# ##                                           )                            (ASLGEN'D)   160
(F1)                      (2S##                                                     )                            (ASLGEN'D)   161
(F1)                      (63C #PAGE: # F.PAGE# ##                                  )                            (ASLGEN'D)   162
(F1)                      (2S##                                                     )                            (ASLGEN'D)   163
(F1)                      (COLUMN-HEADING SECTION ##                                )                            (ASLGEN'D)   164
(F1)                      (70#*# ##                                                 )                            (ASLGEN'D)   165
(F1)                      (#** ORDER # 4B # ITEM # 4B #QUANTITY# 4B #ITEM # 8B      )                            (ASLGEN'D)   166
(F1)                      (#TOTAL# 69C #**# ##                                      )                            (ASLGEN'D)   167
(F1)                      (#** NUMBER# 4B #NUMBER# 4B # ORDERED# 4B #PRICE# 8B      )                            (ASLGEN'D)   168
(F1)                      (# COST# 69C #**# ##                                      )                            (ASLGEN'D)   169
(F1)                      (70#*# ##                                                 )                            (ASLGEN'D)   170
(F1)                      (SUMMARY SECTION ##                                       )                            (ASLGEN'D)   171
(F1)                      (1A COMPUTE ROUNDED STEMP012 = TOTAL T.TOTITMPC#1 * .0725 ##)                           (ASLGEN'D)   172
** MK4FK20  TYPE 1  DEFAULT ATTRIBUTES FOR STEMP01 - FORMAT P MAXIMUM LENGTH    8 SCALE 2.                                *****
(F1)                      (1A COMPUTE ROUNDED STEMP022 = TOTAL T.TOTITMPC#1 + STEMP01 )                           (ASLGEN'D)   173
** MK4FK20  TYPE 1  DEFAULT ATTRIBUTES FOR STEMP02 - FORMAT P MAXIMUM LENGTH    8 SCALE 2.                                *****
(F1)                      (##                                                       )                            (ASLGEN'D)   174
(F1)                      (1A COMPUTE ROUNDED STEMP032 = TOTAL T.FREIGHT#1 + STEMP02 #)                           (ASLGEN'D)   175
** MK4FK20  TYPE 1  DEFAULT ATTRIBUTES FOR STEMP03 - FORMAT P MAXIMUM LENGTH    8 SCALE 2.                                *****
(F1)                      (#                                                        )                            (ASLGEN'D)   176
(F1)                      (1A 3S##                                                  )                            (ASLGEN'D)   177
(F1)                      (1A 40C #SUBTOTAL   # TOTAL T.TOTITMPC#1 ##               )                            (ASLGEN'D)   178
(F1)                      (1A 40C #SALES TAX      # STEMP01($ZZZZZ9.99) ##         )                            (ASLGEN'D)   179
(F1)                      (1A 40C #FREIGHT CHARGE # TOTAL T.FREIGHT#1 ##            )                            (ASLGEN'D)   180
(F1)                      (1A 56C 9#-# ##                                           )                            (ASLGEN'D)   181
(F1)                      (1A 40C #INVOICE TOTAL  # STEMP03($ZZZZZ9.99) ##          )                            (ASLGEN'D)   182
(F1)                      (1A 56C 9#=# ##                                           )                            (ASLGEN'D)   183
```

Figure D-58 Formatted Sectional Report Commands: Computed Summaries (Page 2 of 2)

## Commands to Generate Report with Preprinted Forms

```
                                                *****************************
                                                *  PROC NAME - INVOICE      *
                                                *  INPUT STREAM PROCEDURE   *
                                                *****************************
----------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                             MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ         BACK BR RNT AB      SEQUENCE  LIST
TYPE  DATE         REQUESTOR ID        ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM      73 TO 80   SEQ
----------------------------------------------------------------------------------------------------------------------------------
(ER)                                                                          (N)                             (ASLGEN'D)     9

     INVOICE: PROC                                                                                                          10
     SALEAMT: FIELD P, LEN 5, FLT '$'                                                                                       11
       IF ORDERNO EQ ORDERNO                                                                                                12
          LET T.SALEAMT = ITMPRICE * ITMQTYOR                                                                               13
     ;                                                                                                                      14
       REPORT  CUSTNAME CUSTNO CUSTPH ORDERNO ,                                                                             15
               ITMQTYOR ITEMORD ITMPRICE T.SALEAMT                                                                          16
           GROUP BY CUSTNAME CUSTNO CUSTPH AT LEVEL 1                                                                       17
           ORDER BY CUSTNAME                                                                                                18
           TOTAL T.SALEAMT AT LEVEL 1                                                                                       19
           FORMAT  DETAILSPACING 2 WIDTH 65                                                                                 20
           ITEM CUSTNAME CUSTNO  NOPRINT                                                                                    21
           ITEM CUSTPH  NOPRINT  PICTURE P'(###) ###-####'                                                                  22
      ;                                                                                                                     23
      SECTION PAGETITLE                                                                                                     24
           SKIP 6 LINES                                                                                                     25
          LINE COL(50) F.DATE                                                                                               26
          LINE COL(5) CUSTNAME  SPACES(22) CUSTNO                                                                           27
          LINE COL(5) CUSTPH                                                                                                28
           SKIP 3 LINES                                                                                                     29
      END SECTION                                                                                                           30
      ;                                                                                                                     31
      SECTION DETAIL                                                                                                        32
           SIZE 10 LINES                                                                                                    33
           NEWPAGE AT LEVEL 1                                                                                               34
      END SECTION                                                                                                           35
      ;                                                                                                                     36
      SECTION SUMMARY                                                                                                       37
          SKIP 1 LINES                                                                                                      38
          LINE COL(35) 'TOTAL' TOTAL(T.SALEAMT) AT LEVEL 1                                                                  39
      END SECTION                                                                                                           40
      ;                                                                                                                     41
       END REPORT                                                                                                           42
     ;                                                                                                                      43
       END IF                                                                                                               44
     END PROC                                                                                                               45
----------------------------------------------------------------------------------------------------------------------------------
STMT   FIELD     FIELD FLD DEC OUTPT EDIT      INITIAL          LINE 1 OF        LINE 2 OF                     SEQUENCE  LIST
TYPE   NAME      LNGTH TYP PLC EDIT  LGTH       VALUE         COLUMN HEADING   COLUMN HEADING                  73 TO 80   SEQ
----------------------------------------------------------------------------------------------------------------------------------
(TF)  (SALEAMT ) (  5) (P)      ($  )                                                                         (ASLGEN'D)    46
----------------------------------------------------------------------------------------------------------------------------------
STMT SEQ  LOG CON .....OPERAND-A.....  OPER  ...........OPERAND-B............   ......RESULT......  PARTIAL FIELD  SEQUENCE  LIST
TYPE NO.  LEV CTR QLF FIELD   SEG-LVL  ATN   QLF FIELD   SEG-LVL  (OR CONSTANT)  QLF FIELD  SEG-LVL  STRT NUMB OPR  73 TO 80   SEQ
----------------------------------------------------------------------------------------------------------------------------------
(PR) (100)          (ORDERNO ) 002-2  (EQ)    (ORDERNO ) 002-2                                            (ASLGEN'D)    47
(PR) (101)                            (NS)    (END                              )                         (ASLGEN'D)    48
(PR) (102)          (ITMPRICE) 005-3  (* )    (ITMQTYOR) 005-3                   (T,SALEAMT )              (ASLGEN'D)    49

----------------------------------------------------------------------------------------------------------------------------------
STMT SEQ  LOG CON .....OPERAND-A.....  OPER  ...........OPERAND-B............   ......RESULT......  PARTIAL FIELD  SEQUENCE  LIST
TYPE NO.  LEV CTR QLF FIELD   SEG-LVL  ATN   QLF FIELD   SEG-LVL  (OR CONSTANT)  QLF FIELD  SEG-LVL  STRT NUMB OPR  73 TO 80   SEQ
----------------------------------------------------------------------------------------------------------------------------------
(PR) (103)                            (GO)    (SUB REPT___3                     )                         (ASLGEN'D)    50

                                                *****************************
                                                *  REQUEST NAME - REPT___3  *
                                                *  GENERATED REQUEST        *
                                                *****************************
----------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                             MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ         BACK BR RNT AB      SEQUENCE  LIST
TYPE  DATE         REQUESTOR ID        ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM      73 TO 80   SEQ
----------------------------------------------------------------------------------------------------------------------------------
(ER) (TODAY )                                                                 (S)                             (ASLGEN'D)    51
----------------------------------------------------------------------------------------------------------------------------------
 ST SUM V  8  PG  PG   I SP  MAXIMUM TB COLUMN  DT  PG  STRT LIN SL RH .......SUBFILE........ C E A S R S N      C  SEQUENCE  LIST
 TY RPT S LPI WID HGT  M FRM LPP PGS TO HEADING POS POS PAGE NUM UB PN ENT     NAME   BLK/ FMT H F G T S U D      O  73 TO 80   SEQ
 MP     P              G              LT TYP POS        NUM     ML TD REC             BUF     C C S L T M S      N
----------------------------------------------------------------------------------------------------------------------------------
(E1)    (2)    (065)                                                                                          (ASLGEN'D)    52
----------------------------------------------------------------------------------------------------------------------------------
          SP  Q                E  N  S  D  C  S  T  C  C  M  M  A  M  D Q PCT/RAT      P                PARTIAL  SEQUENCE  LIST
```

Figure D-59     Commands to Generate Report with Preprinted Forms (Page 1 of 2)

```
STMT SEQ  BF  L  FIELD   SEG-   N  P  E  E  T  U  O  U  N  A  I  V  O  E  L  FIELD   RA- C      PICTURE     STRT NO   73 TO 80    SEQ
TYPE NO.  COL F  NAME      LVL  D  R  Q  S  L  B  T  M  T  X  N  G  D  C  F  NAME    TIO T       EDIT       CHR CHR
-------------------------------------------------------------------------------------------------------------------------------------
(R1)           (CUSTNAME) 001-1   (Y) (1)    (1)                                                          (ASLGEN'D)     53
(R1)           (CUSTNO  ) 001-1   (Y)       (1)                                                          (ASLGEN'D)     54
(R1)           (CUSTPH  ) 001-1   (Y)       (1)                                                          (ASLGEN'D)     55
** MK4FW09  TYPE 0  THE EXTENDED EDIT PATTERN FOR FIELD  ,CUSTPH   WILL BE USED - (###) ###-####            .                 *****
(R1)           (ORDERNO ) 002-2                                                                           (ASLGEN'D)     56
(R1)           (ITMQTYOR) 005-3                                                                           (ASLGEN'D)     57
(R1)           (ITEMORD ) 005-3                                                                           (ASLGEN'D)     58
(R1)           (ITMPRICE) 005-3                                                                           (ASLGEN'D)     59
(R1)         (T,SALEAMT )                        (1)                                                      (ASLGEN'D)     60
-------------------------------------------------------------------------------------------------------------------------------------
STMT  SEQ                                                                                                  SEQUENCE    LIST
TYPE  NO.                                              TEXT                                                73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------------
(F1)                      (6S##                                                          )                 (ASLGEN'D)     61
(F1)                      (50C F.DATE# ##                                                )                 (ASLGEN'D)     62
(F1)                      (5C .CUSTNAME# 22B .CUSTNO# ##                                 )                 (ASLGEN'D)     63
(F1)                      (5C .CUSTPH# ##                                                )                 (ASLGEN'D)     64
(F1)                      (3S##                                                          )                 (ASLGEN'D)     65
(F1)                      (10D1 ##                                                       )                 (ASLGEN'D)     66
(F1)                      (1S##                                                          )                 (ASLGEN'D)     67
(F1)                      (1A 35C #TOTAL# TOTAL T.SALEAMT#1 ##                           )                 (ASLGEN'D)     68
```

Figure D-59 Commands to Generate Report with Preprinted Forms (Page 2 of 2)

## Commands to Generate Report Using Dynamic Report Line Modification

```
                                                  ****************************
                                                  *  PROC NAME - XREP        *
                                                  *  INPUT STREAM PROCEDURE  *
                                                  ****************************
-------------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                              MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB       SEQUENCE    LIST
TYPE  DATE          REQUESTOR ID         ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM      73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------------
(ER)                                                                          (N)                              (ASLGEN'D)      9

    XREP: PROC                                                                                                                10
    ; dynamic report line modification used to modify report                                                                   11
    ;                                  after summaries are taken                                                              12
    IF PF(ITEMORD 4 1) GT 'H' THEN                                                                                            13
        LET T.DEPT = 'B'                                                                                                      14
        LET T.GROSS_A = ITMPRICE * QTYSHIP                                                                                    15
      ELSE                                                                                                                    16
        LET T.DEPT = 'A'                                                                                                      17
        LET T.GROSS_B = ITMPRICE * QTYSHIP                                                                                    18
     END IF                                                                                                                   19
     LET T.RATIO = 0                                                                                                          20
     ; report without Dynamic Report Line Modification                                                                        21
     REPORT  T.DEPT T.GROSS_A T.GROSS_B T.RATIO  SUMMARYONLY                                                                  22
        FORMAT WIDTH 70                                                                                                       23
        ORDER BY T.DEPT                                                                                                       24
        ORDER BY T.DEPT                                                                                                       25
        GROUP BY T.DEPT                                                                                                       26
        TOTAL T.GROSS_A T.GROSS_B AT LEVEL G                                                                                  27
        TITLE 'COMPARISION OF DEPARTMENT GROSS SALES'                                                                         28
     END REPORT                                                                                                               29
     ; final result with Dynamic Report Line Modification                                                                     30
     REPORT  T.DEPT T.GROSS_A T.GROSS_B T.RATIO  SUMMARYONLY                                                                  31
        FORMAT WIDTH 70                                                                                                       32
        ORDER BY T.DEPT                                                                                                       33
        ORDER BY T.DEPT                                                                                                       34
        GROUP BY T.DEPT                                                                                                       35
        TOTAL T.GROSS_A T.GROSS_B AT LEVEL G                                                                                  36
        TITLE 'COMPARISION OF DEPARTMENT GROSS SALES'                                                                         37
        XREP 'TITLE LINENO=5'                                                                                                 38
        XREP 'RATIO A=(024,9,999),B=(037,999),C=(050,99.99)'                                                                  39
        XREP 'ROUND A=(050,99.99)'                                                                                            40
     END REPORT                                                                                                               41
    END PROC                                                                                                                  42
-------------------------------------------------------------------------------------------------------------------------------------
STMT SEQ  LOG CON .....OPERAND-A.....  OPER  .............OPERAND-B.............   ......RESULT......  PARTIAL FIELD  SEQUENCE   LIST
TYPE NO.  LEV CTR QLF FIELD   SEG-LVL  ATN   QLF FIELD   SEG-LVL  (OR CONSTANT)   QLF FIELD  SEG-LVL  STRT NUMB OPR  73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------------
(PR) (100)          (ITEMORD ) 005-3  (GT)    (C,H                          )                      ( 4) ( 1) (A) (ASLGEN'D)     43
(PR) (101)                            (NS)      (105                          )                                    (ASLGEN'D)     44
(PR) (102)                            (R )    (C,B                          ) (T,DEPT    )                       (ASLGEN'D)     45
** MK4DI02  TYPE 0  DEFINITION OF TEMPORARY FIELD DEPT     - FORMAT C  LENGTH   1  DECIMALS 0 EDIT LENGTH   1 .                 *****
```

Figure D-60 Commands to Generate Report Using Dynamic Report Line Modification (Page 1 of 2)

```
(PR) (103)          (ITMPRICE) 005-3  (* )     (QTYSHIP ) 004-4                (T,GROSS_A )                          (ASLGEN'D)    46
** MK4DI02  TYPE 0  DEFINITION OF TEMPORARY FIELD GROSS_A  - FORMAT Z  LENGTH   6  DECIMALS 2 EDIT LENGTH   9 .                 *****
(PR) (104)                            (GO)     (107                           )                                    (ASLGEN'D)    47
(PR) (105)                            (R )     (C,A                           ) (T,DEPT    )                       (ASLGEN'D)    48
(PR) (106)          (ITMPRICE) 005-3  (* )     (QTYSHIP ) 004-4                (T,GROSS_B )                          (ASLGEN'D)    49
** MK4DI02  TYPE 0  DEFINITION OF TEMPORARY FIELD GROSS_B  - FORMAT Z  LENGTH   6  DECIMALS 2 EDIT LENGTH   9 .                 *****
(PR) (107)                            (R )     (D,0                           ) (T,RATIO   )                       (ASLGEN'D)    50
** MK4DI02  TYPE 0  DEFINITION OF TEMPORARY FIELD RATIO    - FORMAT P  LENGTH   8  DECIMALS 0 EDIT LENGTH  20 .                 *****
(PR) (108)                            (GO)     (SUB REPT___5                  )                                    (ASLGEN'D)    51
(PR) (109)                            (GO)     (SUB REPT___6                  )                                    (ASLGEN'D)    52

                                                ******************************
                                                *  REQUEST NAME - REPT___5   *
                                                *  GENERATED REQUEST         *
                                                ******************************

-------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                                MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB      SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID           ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM      73 TO 80   SEQ
-------------------------------------------------------------------------------------------------------------------------------
(ER) (TODAY )                                                                    (S)                                  (ASLGEN'D)    53
-------------------------------------------------------------------------------------------------------------------------------
 ST SUM V  8  PG  PG   I SP  MAXIMUM TB COLUMN  DT  PG  STRT LIN SL RH .......SUBFILE..... C E A S R S N      C  SEQUENCE   LIST
 TY RPT S LPI WID HGT  M FRM LPP PGS TO HEADING POS POS PAGE NUM UB PN ENT    NAME   BLK/ FMT H F G T S U D      O  73 TO 80   SEQ
 MP     P            G            LT TYP POS         NUM      ML TD REC           BUF      C C S L T M S      N
-------------------------------------------------------------------------------------------------------------------------------
(E1) (Y)      (070)                                                                                                  (ASLGEN'D)    54
-------------------------------------------------------------------------------------------------------------------------------
         SP  Q              E N S  D  C  S  T  C  C  M  M  A  M  D Q PCT/RAT     P                PARTIAL   SEQUENCE   LIST
STMT SEQ BF  L  FIELD  SEG- N P E  E  T  U  O  U  N  A  I  V  O  E L  FIELD   RA- C     PICTURE    STRT NO   73 TO 80   SEQ
TYPE NO. COL F  NAME     LVL D R Q  S  L  B  T  M  T  X  N  G  D  C F  NAME    TIO T      EDIT     CHR CHR
-------------------------------------------------------------------------------------------------------------------------------
(R1)         (T,DEPT    )          (2)   (1)                                                                       (ASLGEN'D)    55
(R1)         (T,GROSS_A )                      (G)                                                                  (ASLGEN'D)    56
(R1)         (T,GROSS_B )                      (G)                                                                  (ASLGEN'D)    57
(R1)         (T,RATIO   )                                                                                           (ASLGEN'D)    58
-------------------------------------------------------------------------------------------------------------------------------
STMT  SEQ                                                                                                          SEQUENCE   LIST
TYPE  NO.                                        TEXT                                                              73 TO 80   SEQ
-------------------------------------------------------------------------------------------------------------------------------
(T1)  (101)              (COMPARISION OF DEPARTMENT GROSS SALES#                 )                                  (ASLGEN'D)    59

                                                ******************************
                                                *  REQUEST NAME - REPT___6   *
                                                *  GENERATED REQUEST         *
                                                ******************************

-------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                                MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB      SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID           ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM      73 TO 80   SEQ
-------------------------------------------------------------------------------------------------------------------------------
(ER) (TODAY )                                                                    (S)                                  (ASLGEN'D)    60
-------------------------------------------------------------------------------------------------------------------------------
 ST SUM V  8  PG  PG   I SP  MAXIMUM TB COLUMN  DT  PG  STRT LIN SL RH .......SUBFILE..... C E A S R S N      C  SEQUENCE   LIST
 TY RPT S LPI WID HGT  M FRM LPP PGS TO HEADING POS POS PAGE NUM UB PN ENT    NAME   BLK/ FMT H F G T S U D      O  73 TO 80   SEQ
 MP     P            G            LT TYP POS         NUM      ML TD REC           BUF      C C S L T M S      N
-------------------------------------------------------------------------------------------------------------------------------
(E1) (Y)      (070)                                                                                                  (ASLGEN'D)    61
-------------------------------------------------------------------------------------------------------------------------------
STMT  SEQ                                                                                                          SEQUENCE   LIST
TYPE  NO.                                        TEXT                                                              73 TO 80   SEQ
-------------------------------------------------------------------------------------------------------------------------------
(P1)  (000)              (MARKXREP#                                             )                                  (ASLGEN'D)    62
(P1)  (001)              (TITLE LINENO=5#                                       )                                  (ASLGEN'D)    63
(P1)  (002)              (RATIO A=(024,9,999),B=(037,999),C=(050,99.99)#        )                                  (ASLGEN'D)    64
(P1)  (003)              (ROUND A=(050,99.99)#                                  )                                  (ASLGEN'D)    65
-------------------------------------------------------------------------------------------------------------------------------
         SP  Q              E N S  D  C  S  T  C  C  M  M  A  M  D Q PCT/RAT     P                PARTIAL   SEQUENCE   LIST
STMT SEQ BF  L  FIELD  SEG- N P E  E  T  U  O  U  N  A  I  V  O  E L  FIELD   RA- C     PICTURE    STRT NO   73 TO 80   SEQ
TYPE NO. COL F  NAME     LVL D R Q  S  L  B  T  M  T  X  N  G  D  C F  NAME    TIO T      EDIT     CHR CHR
-------------------------------------------------------------------------------------------------------------------------------
(R1)         (T,DEPT    )          (2)   (1)                                                                       (ASLGEN'D)    66
(R1)         (T,GROSS_A )                      (G)                                                                  (ASLGEN'D)    67
(R1)         (T,GROSS_B )                      (G)                                                                  (ASLGEN'D)    68
(R1)         (T,RATIO   )                                                                                           (ASLGEN'D)    69
-------------------------------------------------------------------------------------------------------------------------------
STMT  SEQ                                                                                                          SEQUENCE   LIST
TYPE  NO.                                        TEXT                                                              73 TO 80   SEQ
-------------------------------------------------------------------------------------------------------------------------------
(T1)  (101)              (COMPARISION OF DEPARTMENT GROSS SALES#                 )                                  (ASLGEN'D)    70
```

Figure D-60 Commands to Generate Report Using Dynamic Report Line Modification (Page 2 of 2)

# Advanced Transaction Processing

## Definition Run

### Banner Page

```
FEB 11, 2002  14.27.06                                                                                    PAGE      1

                                        ******************************************
                                        * VISION:Builder   4400  (OS/390 - 14.0)  *
                                        *          COPYRIGHT (C) 2002            *
                                        * COMPUTER ASSOCIATES INTERNATIONAL, INC. *
                                        ******************************************

VISION:Builder JCR1 14.0 020802            ENABLED FOR IBM LANGUAGE ENVIRONMENT                BUILD STAMP = 102039,18:29:48.
                                         COMLIB  MODIFICATIONS (PTFs, APARs) INSTALLED
                                      201,202,203,204,205,206,207,208,209,210,212,213,214,215,216

=============================================================================================================================
=                                                                                                                           =
=                  I N S T A L L A T I O N   P A R A M E T E R S   ( M 4 P A R A M S , M A R K L I B P )                    =
=                                                                                                                           =
=  SYSTEM DELIMITER:   #          PAGE HEIGHT:        66        M4LIST WIDTH:      132       DEF WIDTH OF PAGE:  0        =
=                                                                                                                           =
=  AUTO GRAND:         N          HEADING CHAR:       -         SUBTITLE REPEAT:   N         INVALID FIELD:      *        =
=                                                                                                                           =
=  MISSING FIELD:      -          NON-EDIT FIELD:     +         PERCENT CHAR:      %         LEFT SEPARATOR:     (        =
=                                                                                                                           =
=  RIGHT SEPARATOR:    )          SINGLE SEPARATOR:   ,         SOURCE SPACING:    1         PRINT MESSAGES:     Y        =
=                                                                                                                           =
=  CONSOLE MESSAGES:   N          M4REPO BLOCKSIZE:   4,096     INPUT I/O BUFFERS: 2         OUTPUT I/O BUFFERS: 1        =
=                                                                                                                           =
=  SNGL-STEP STORAGE:  8,192      SNGL-STEP SORTSIZE: 524,288   DIGIT SELECT CHAR: 9         ZERO SUPPRESS CHAR: Z        =
=                                                                                                                           =
=  CURRENCY CHAR:      $          PLUS CHAR:          +         MINUS CHAR:        -         CHECK PROTECT CHAR: *        =
=                                                                                                                           =
=  DECIMAL CHAR:       .          GROUPING CHAR:      ,         PRIMARY PLOT CHAR: X         SECONDARY PLOT CHAR:*        =
=                                                                                                                           =
=  FIT PLOT CHAR:      .          HORIZONTAL AXIS:    _         HORIZONTAL HASH:   |         VERTICAL AXIS:     |        =
=                                                                                                                           =
=  VERTICAL HASH:      -          MINUTES/HOUR:       60        SECONDS/MINUTE:    60        TIME DELIMITER:     HH:MM:SS =
=                                                                                                                           =
=  DATE FORMAT:      MM DD, YYYY  TODAY FORMAT+DELIM: MM/DD/YY   ISDATE DELIMITER:  YYYY-MM-DD JULIAN DELIMITER:  YY.DDD    =
=                                                                                                                           =
=  SORT PROGRAM CODE:  2          MINCORE VALUE:      12 K      ALT M4LIST WIDTH:  264       ALT DEF W/OF PAGE:  0        =
=                                                                                                                           =
=  MAX LINES OF TRACE: 1,024      ITEM TRACKING:      0         SUPPRESS NDS REPT?: N        DEFAULT MAXGETMN:   1,024 K  =
=                                                                                                                           =
=  CONDITION CODE 1:   0          CONDITION CODE 2:   4         CONDITION CODE 3:  8         CONDITION CODE 4:   16       =
=                                                                                                                           =
=  ISAM INDEX INCORE:  N          M4LIB BLKG FACTOR:  0         M4LIB RESERVE:     0         M4LIB COMPONENT:    C4.5 B/V =
=                                                                                                                           =
=  DEFAULT F/P AMODE:  24         M4LIB COMPRESSION?: N         M4LIB COMPRESS MIN: 507                                   =
=                                                                                                                           =
=============================================================================================================================

FEB 11, 2002  14.27.06                                                                                    PAGE      2
-----------------------------------------------------------------------------------------------------------------------------
   RUN     STMT    FILE    O N S D U T R A B S D S L  RB  R S M R S          START          END           SEQUENCE   LIST
   NAME    TYPE    NAME    L E E I P R E U U R L C S  PL  J O O F S          SEARCH         SEARCH        73 TO 80   SEQ
                           D W Q R D N P D F T M N T  TK  T P P O R
-----------------------------------------------------------------------------------------------------------------------------
(DEFINE  ) (RC)                                                                                                          1
```

Figure D-61 Definition Run: Banner Page

### CUSTMAST File Definition

```
FEB 11, 2002  14.27.06                                                                                    PAGE      3
** CLSAA00  TYPE 0  DEFINITION CUSTMAST IS BEING PROCESSED.                                                      *****
-----------------------------------------------------------------------------------------------------------------------------
   FILE    STMT    FILE    DELETE GLOSSARY RECORD RECORD RECORDS BUFFER MAPPED     MAPPING          MAPPED       SEQUENCE   LIST
   NAME    TYPE    IDENT    FILE   LEVEL   FORMAT  SIZE  PER BLK  SIZE  FILE    INIT     TERM    RECSIZ       73 TO 80   SEQ
-----------------------------------------------------------------------------------------------------------------------------
(CUSTMAST) (FD)             (R)             (V)                ( 2500)                                                   2
-----------------------------------------------------------------------------------------------------------------------------
   FILE    STMT  SEGMENT  DEL SGMT VRT            SUP SEG  KEY              AUTH          EXTERNAL          SEQUENCE   LIST
   NAME    TYPE  NAME     ETE NMBR SEG            DPL KEY  CHAR             ID          TABLE NAME          73 TO 80   SEQ
-----------------------------------------------------------------------------------------------------------------------------
```

Figure D-62 Definition Run: CUSTMAST File Definition (Page 1 of 7)

```
(CUSTMAST) (LS) (CUSTOMER)    ( 1)                                                                                      3
--------------------------------------------------------------------------------------------------------------------------
   FILE     STMT   FIELD   DEL SGMT LEV FIELD  FIELD FLD SEG FLD DEC CNT OCRS OUTP EDIT      DATA BASE NAME/        SEQUENCE   LIST
   NAME     TYPE   NAME    ETE NMBR NUM LOCATN LNGTH TYP KEY RND PLC SEG NTIM EDIT LNGH      COLUMN HEADING         73 TO 80    SEQ
--------------------------------------------------------------------------------------------------------------------------
(CUSTMAST) (L0) (CUSTNO  )    ( 1) (1) (   1) (  5) (C) (1)                                                             4
(CUSTMAST) (L1) (CUSTNO  )                                               (CUSTOMER                  )                   5
(CUSTMAST) (L2) (CUSTNO  )                                               (NUMBER                    )                   6
(CUSTMAST) (L0) (CUSTNAME)    ( 1) (1) (   6) ( 30) (C)                                                                 7
(CUSTMAST) (L1) (CUSTNAME)                                               (CUSTOMER                  )                   8
(CUSTMAST) (L2) (CUSTNAME)                                               (NAME                      )                   9
(CUSTMAST) (L0) (CUSTPH  )    ( 1) (1) (  36) ( 10) (C)                                                                10
(CUSTMAST) (L1) (CUSTPH  )                                               (CUSTOMER                  )                  11
(CUSTMAST) (L2) (CUSTPH  )                                               (PHONE                     )                  12
(CUSTMAST) (L0) (CUSTADDR)    ( 1) (1) (  46) ( 25) (C)                                                                13
(CUSTMAST) (L1) (CUSTADDR)                                               (CUSTOMER                  )                  14
(CUSTMAST) (L2) (CUSTADDR)                                               (ADDRESS                   )                  15
(CUSTMAST) (L0) (CITY    )    ( 1) (1) (  71) ( 23) (C)                                                                16
(CUSTMAST) (L0) (STATE   )    ( 1) (1) (  94) (  2) (C)                                                                17
(CUSTMAST) (L1) (STATE   )                                                                                             18
(CUSTMAST) (L2) (STATE   )                                               (STATE                     )                  19
(CUSTMAST) (L0) (CUSTZIP )    ( 1) (1) (  96) (  5) (C)                                                                20
(CUSTMAST) (L1) (CUSTZIP )                                               (ZIP                       )                  21
(CUSTMAST) (L2) (CUSTZIP )                                               (CODE                      )                  22
(CUSTMAST) (L0) (SEG20CNT)    ( 1) (1) ( 101) (  2) (F)         ( 2)                                                   23
(CUSTMAST) (L0) (SEG60CNT)    ( 1) (1) ( 103) (  2) (F)         ( 6)                                                   24
--------------------------------------------------------------------------------------------------------------------------
   FILE     STMT  SEGMENT  DEL SGMT VRT             SUP SEG  KEY                 AUTH         EXTERNAL          SEQUENCE   LIST
   NAME     TYPE   NAME    ETE NMBR SEG             DPL KEY CHAR                  ID         TABLE NAME         73 TO 80    SEQ
--------------------------------------------------------------------------------------------------------------------------
(CUSTMAST) (LS) (ORDER   )    ( 2)                      (A)                                                            25
--------------------------------------------------------------------------------------------------------------------------
   FILE     STMT   FIELD   DEL SGMT LEV FIELD  FIELD FLD SEG FLD DEC CNT OCRS OUTP EDIT      DATA BASE NAME/        SEQUENCE   LIST
   NAME     TYPE   NAME    ETE NMBR NUM LOCATN LNGTH TYP KEY RND PLC SEG NTIM EDIT LNGH      COLUMN HEADING         73 TO 80    SEQ
--------------------------------------------------------------------------------------------------------------------------
(CUSTMAST) (L0) (ORDERNO )    ( 2) (2) (   1) (  5) (C) (1)                                                            26
(CUSTMAST) (L1) (ORDERNO )                                               (ORDER                     )                  27
(CUSTMAST) (L2) (ORDERNO )                                               (NUMBER                    )                  28
(CUSTMAST) (L0) (ORDRDATE)    ( 2) (2) (   8) (  6) (C)                                                                29
(CUSTMAST) (L1) (ORDRDATE)                                               (ORDER                     )                  30
(CUSTMAST) (L2) (ORDRDATE)                                               (DATE                      )                  31
(CUSTMAST) (L0) (ORPERSON)    ( 2) (2) (  14) (  3) (C)                                                                32
(CUSTMAST) (L1) (ORPERSON)                                               (SALES                     )                  33
(CUSTMAST) (L2) (ORPERSON)                                               (PERSON                    )                  34
(CUSTMAST) (L0) (ORDPONUM)    ( 2) (2) (  17) (  5) (C)                                                                35
(CUSTMAST) (L1) (ORDPONUM)                                               (PURCH. ORDER              )                  36
(CUSTMAST) (L2) (ORDPONUM)                                               (NUMBER                    )                  37
(CUSTMAST) (L0) (ORDDUDAT)    ( 2) (2) (  22) (  6) (C)                                                                38
FEB 11, 2002  14.27.06                                                                                          PAGE    4
--------------------------------------------------------------------------------------------------------------------------
   FILE     STMT   FIELD   DEL SGMT LEV FIELD  FIELD FLD SEG FLD DEC CNT OCRS OUTP EDIT      DATA BASE NAME/        SEQUENCE   LIST
   NAME     TYPE   NAME    ETE NMBR NUM LOCATN LNGTH TYP KEY RND PLC SEG NTIM EDIT LNGH      COLUMN HEADING         73 TO 80    SEQ
--------------------------------------------------------------------------------------------------------------------------
(CUSTMAST) (L1) (ORDDUDAT)                                               (DUE                       )                  39
(CUSTMAST) (L2) (ORDDUDAT)                                               (DATE                      )                  40
(CUSTMAST) (L0) (ORDMONTH)    ( 2) (2) (   8) (  2) (C)                                                                41
(CUSTMAST) (L1) (ORDMONTH)                                               (ORDER                     )                  42
(CUSTMAST) (L2) (ORDMONTH)                                               (MONTH                     )                  43
(CUSTMAST) (L0) (ORDERDAY)    ( 2) (2) (  10) (  2) (C)                                                                44
(CUSTMAST) (L1) (ORDERDAY)                                               (ORDER                     )                  45
(CUSTMAST) (L2) (ORDERDAY)                                               (DAY                       )                  46
(CUSTMAST) (L0) (ORDRYEAR)    ( 2) (2) (  12) (  2) (C)                                                                47
(CUSTMAST) (L1) (ORDRYEAR)                                               (ORDER                     )                  48
(CUSTMAST) (L2) (ORDRYEAR)                                               (YEAR                      )                  49
(CUSTMAST) (L0) (ORDINVGN)    ( 2) (2) (  28) (  1) (C)                                                                50
(CUSTMAST) (L1) (ORDINVGN)                                               (INVOICE                   )                  51
(CUSTMAST) (L2) (ORDINVGN)                                               (GENERATED?                )                  52
(CUSTMAST) (L0) (ORDCMPLT)    ( 2) (2) (  29) (  1) (C)                                                                53
(CUSTMAST) (L1) (ORDCMPLT)                                               (ORDER                     )                  54
(CUSTMAST) (L2) (ORDCMPLT)                                               (COMPLETED?                )                  55
(CUSTMAST) (L0) (SEG30CNT)    ( 2) (2) (  40) (  2) (F)         ( 3)                                                   56
(CUSTMAST) (L0) (SEG50CNT)    ( 2) (2) (  42) (  2) (F)         ( 5)                                                   57
(CUSTMAST) (L0) (DUEMONTH)    ( 2) (2) (  22) (  2) (C)                                                                58
(CUSTMAST) (L1) (DUEMONTH)                                               (MONTH                     )                  59
(CUSTMAST) (L2) (DUEMONTH)                                               (DUE                       )                  60
(CUSTMAST) (L0) (DUEDAY  )    ( 2) (2) (  24) (  2) (C)                                                                61
(CUSTMAST) (L1) (DUEDAY  )                                               (DAY                       )                  62
(CUSTMAST) (L2) (DUEDAY  )                                               (DUE                       )                  63
(CUSTMAST) (L0) (DUEYEAR )    ( 2) (2) (  26) (  2) (C)                                                                64
(CUSTMAST) (L1) (DUEYEAR )                                               (YEAR                      )                  65
(CUSTMAST) (L2) (DUEYEAR )                                               (DUE                       )                  66
--------------------------------------------------------------------------------------------------------------------------
   FILE     STMT  SEGMENT  DEL SGMT VRT             SUP SEG  KEY                 AUTH         EXTERNAL          SEQUENCE   LIST
   NAME     TYPE   NAME    ETE NMBR SEG             DPL KEY CHAR                  ID         TABLE NAME         73 TO 80    SEQ
--------------------------------------------------------------------------------------------------------------------------
(CUSTMAST) (LS) (SHIPINV )    ( 3)                      (A)                                                            67
```

Figure D-62    Definition Run: CUSTMAST File Definition (Page 2 of 7)

```
------------------------------------------------------------------------------------------------------------------------------------
  FILE    STMT   FIELD   DEL SGMT LEV FIELD  FIELD FLD SEG FLD DEC CNT OCRS OUTP EDIT       DATA BASE NAME/           SEQUENCE  LIST
  NAME    TYPE   NAME    ETE NMBR NUM LOCATN LNGTH TYP KEY RND PLC SEG NTIM EDIT LNGH       COLUMN HEADING            73 TO 80   SEQ
------------------------------------------------------------------------------------------------------------------------------------
(CUSTMAST) (L0) (SHIPNO  )   ( 3)  (3)  (    1)  (  4)  (C)  (1)                                                                 68
(CUSTMAST) (L1) (SHIPNO  )                                                           (SHIPPING                 )                 69
(CUSTMAST) (L2) (SHIPNO  )                                                           (NUMBER                   )                 70
(CUSTMAST) (L0) (SHIPDATE)   ( 3)  (3)  (    5)  (  6)  (C)                                                                      71
(CUSTMAST) (L1) (SHIPDATE)                                                           (DATE                     )                 72
(CUSTMAST) (L2) (SHIPDATE)                                                           (SHIPPED                  )                 73
(CUSTMAST) (L0) (SHIPPER )   ( 3)  (3)  (   11)  (  3)  (C)                                                                      74
(CUSTMAST) (L1) (SHIPPER )                                                           (SHIPPER                  )                 75
(CUSTMAST) (L0) (FRTCOST )   ( 3)  (3)  (   23)  (  4)  (P)        (2)         ($  )                                             76
(CUSTMAST) (L1) (FRTCOST )                                                           (FREIGHT                  )                 77
(CUSTMAST) (L2) (FRTCOST )                                                           (COST                     )                 78
(CUSTMAST) (L0) (INVNO   )   ( 3)  (3)  (   41)  (  5)  (C)                                                                      79
(CUSTMAST) (L1) (INVNO   )                                                           (INVOICE                  )                 80
(CUSTMAST) (L2) (INVNO   )                                                           (NUMBER                   )                 81
(CUSTMAST) (L0) (INVDATE )   ( 3)  (3)  (   48)  (  6)  (C)                                                                      82
(CUSTMAST) (L1) (INVDATE )                                                           (INVOICE                  )                 83
(CUSTMAST) (L2) (INVDATE )                                                           (DATE                     )                 84
FEB 11, 2002  14.27.06                                                                                                   PAGE     5
------------------------------------------------------------------------------------------------------------------------------------
  FILE    STMT   FIELD   DEL SGMT LEV FIELD  FIELD FLD SEG FLD DEC CNT OCRS OUTP EDIT       DATA BASE NAME/           SEQUENCE  LIST
  NAME    TYPE   NAME    ETE NMBR NUM LOCATN LNGTH TYP KEY RND PLC SEG NTIM EDIT LNGH       COLUMN HEADING            73 TO 80   SEQ
------------------------------------------------------------------------------------------------------------------------------------
(CUSTMAST) (L0) (INAMT   )   ( 3)  (3)  (   54)  (  4)  (P)        (2)         ($  )                                             85
(CUSTMAST) (L1) (INAMT   )                                                           (INVOICE                  )                 86
(CUSTMAST) (L2) (INAMT   )                                                           (AMOUNT                   )                 87
(CUSTMAST) (L0) (PAYDATE )   ( 3)  (3)  (   70)  (  6)  (C)                                                                      88
(CUSTMAST) (L1) (PAYDATE )                                                           (PAYMENT                  )                 89
(CUSTMAST) (L2) (PAYDATE )                                                           (DATE                     )                 90
(CUSTMAST) (L0) (CHECKNO )   ( 3)  (3)  (   76)  (  7)  (C)                                                                      91
(CUSTMAST) (L1) (CHECKNO )                                                           (CHECK                    )                 92
(CUSTMAST) (L2) (CHECKNO )                                                           (NUMBER                   )                 93
(CUSTMAST) (L0) (PAIDDATE)   ( 3)  (3)  (   83)  (  6)  (C)                                                                      94
(CUSTMAST) (L1) (PAIDDATE)                                                           (DATE OF                  )                 95
(CUSTMAST) (L2) (PAIDDATE)                                                           (PAYMENT                  )                 96
(CUSTMAST) (L0) (SEG40CNT)   ( 3)  (3)  (   99)  (  2)  (F)            ( 4)                                                      97
------------------------------------------------------------------------------------------------------------------------------------
  FILE    STMT  SEGMENT  DEL SGMT VRT              SUP SEG  KEY                AUTH          EXTERNAL             SEQUENCE  LIST
  NAME    TYPE   NAME    ETE NMBR SEG              DPL KEY CHAR                 ID          TABLE NAME            73 TO 80   SEQ
------------------------------------------------------------------------------------------------------------------------------------
(CUSTMAST) (LS) (ITEMSHIP)   ( 4)                      (A)                                                                       98
------------------------------------------------------------------------------------------------------------------------------------
  FILE    STMT   FIELD   DEL SGMT LEV FIELD  FIELD FLD SEG FLD DEC CNT OCRS OUTP EDIT       DATA BASE NAME/           SEQUENCE  LIST
  NAME    TYPE   NAME    ETE NMBR NUM LOCATN LNGTH TYP KEY RND PLC SEG NTIM EDIT LNGH       COLUMN HEADING            73 TO 80   SEQ
------------------------------------------------------------------------------------------------------------------------------------
(CUSTMAST) (L0) (ITEMSHIP)   ( 4)  (4)  (    1)  (  7)  (C)  (1)                                                                 99
(CUSTMAST) (L1) (ITEMSHIP)                                                           (STOCK NO.                )                100
(CUSTMAST) (L2) (ITEMSHIP)                                                           (SHIPPED                  )                101
(CUSTMAST) (L0) (QTYSHIP )   ( 4)  (4)  (    8)  (  7)  (Z)                                                                     102
(CUSTMAST) (L1) (QTYSHIP )                                                           (QUANTITY                 )                103
(CUSTMAST) (L2) (QTYSHIP )                                                           (SHIPPED                  )                104
(CUSTMAST) (L0) (SEG40FIL)   ( 4)  (4)  (   15)  ( 10)  (C)                                                                     105
------------------------------------------------------------------------------------------------------------------------------------
  FILE    STMT  SEGMENT  DEL SGMT VRT              SUP SEG  KEY                AUTH          EXTERNAL             SEQUENCE  LIST
  NAME    TYPE   NAME    ETE NMBR SEG              DPL KEY CHAR                 ID          TABLE NAME            73 TO 80   SEQ
------------------------------------------------------------------------------------------------------------------------------------
(CUSTMAST) (LS) (ITEMORD )   ( 5)                      (A)                                                                      106
------------------------------------------------------------------------------------------------------------------------------------
  FILE    STMT   FIELD   DEL SGMT LEV FIELD  FIELD FLD SEG FLD DEC CNT OCRS OUTP EDIT       DATA BASE NAME/           SEQUENCE  LIST
  NAME    TYPE   NAME    ETE NMBR NUM LOCATN LNGTH TYP KEY RND PLC SEG NTIM EDIT LNGH       COLUMN HEADING            73 TO 80   SEQ
------------------------------------------------------------------------------------------------------------------------------------
(CUSTMAST) (L0) (ITEMORD )   ( 5)  (3)  (    1)  (  7)  (C)  (1)                                                                107
(CUSTMAST) (L1) (ITEMORD )                                                           (STOCK NO.                )                108
(CUSTMAST) (L2) (ITEMORD )                                                           (ORDERED                  )                109
(CUSTMAST) (L0) (ITEMNAME)   ( 5)  (3)  (    8)  ( 20)  (C)                                                                     110
(CUSTMAST) (L1) (ITEMNAME)                                                           (DESCRIPTION              )                111
(CUSTMAST) (L0) (ITMPRICE)   ( 5)  (3)  (   28)  (  6)  (Z)        (2)         ($  )                                            112
(CUSTMAST) (L1) (ITMPRICE)                                                           (PRICE                    )                113
(CUSTMAST) (L0) (ITMQTYOR)   ( 5)  (3)  (   34)  (  7)  (Z)                                                                     114
(CUSTMAST) (L1) (ITMQTYOR)                                                           (ORDER                    )                115
(CUSTMAST) (L2) (ITMQTYOR)                                                           (QUANTITY                 )                116
(CUSTMAST) (L0) (QTYBKORD)   ( 5)  (3)  (   41)  (  7)  (Z)                                                                     117
(CUSTMAST) (L1) (QTYBKORD)                                                           (QUANTITY                 )                118
(CUSTMAST) (L2) (QTYBKORD)                                                           (BACKORDERED              )                119
(CUSTMAST) (L0) (SEG50FIL)   ( 5)  (3)  (   48)  ( 10)  (C)                                                                     120
FEB 11, 2002  14.27.06                                                                                                   PAGE     6
------------------------------------------------------------------------------------------------------------------------------------
  FILE    STMT  SEGMENT  DEL SGMT VRT              SUP SEG  KEY                AUTH          EXTERNAL             SEQUENCE  LIST
  NAME    TYPE   NAME    ETE NMBR SEG              DPL KEY CHAR                 ID          TABLE NAME            73 TO 80   SEQ
------------------------------------------------------------------------------------------------------------------------------------
(CUSTMAST) (LS) (INSTALL )   ( 6)                      (A)                                                                      121
------------------------------------------------------------------------------------------------------------------------------------
  FILE    STMT   FIELD   DEL SGMT LEV FIELD  FIELD FLD SEG FLD DEC CNT OCRS OUTP EDIT       DATA BASE NAME/           SEQUENCE  LIST
```

Figure D-62    Definition Run: CUSTMAST File Definition (Page 3 of 7)

```
  NAME    TYPE   NAME    ETE NMBR NUM LOCATN LNGTH TYP KEY RND PLC SEG NTIM EDIT LNGH      COLUMN HEADING          73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------------
(CUSTMAST) (L0) (INSTNO  )    ( 6) (2) (   1) (  4) (C) (1)                                                                    122
(CUSTMAST) (L1) (INSTNO  )                                                                  (INSTALLATION              )        123
(CUSTMAST) (L2) (INSTNO  )                                                                  (NUMBER                    )        124
(CUSTMAST) (L0) (INSTCONT)    ( 6) (2) (   5) (  3) (C)                                                                    125
(CUSTMAST) (L1) (INSTCONT)                                                                  (INSTALLATION              )        126
(CUSTMAST) (L2) (INSTCONT)                                                                  (CONTACT                   )        127
(CUSTMAST) (L0) (INSTOPS )    ( 6) (2) (   8) (  8) (C)                                                                    128
(CUSTMAST) (L1) (INSTOPS )                                                                  (OPERATING                 )        129
(CUSTMAST) (L2) (INSTOPS )                                                                  (SYSTEM                    )        130
(CUSTMAST) (L0) (INSTOSRL)    ( 6) (2) (  16) (  2) (C)                                                                    131
(CUSTMAST) (L1) (INSTOSRL)                                                                  (OP SYSTEM                 )        132
(CUSTMAST) (L2) (INSTOSRL)                                                                  (RELEASE                   )        133
(CUSTMAST) (L0) (INPRODID)    ( 6) (2) (  18) (  2) (C)                                                                    134
(CUSTMAST) (L1) (INPRODID)                                                                  (PRODUCT                   )        135
(CUSTMAST) (L2) (INPRODID)                                                                  (RELEASE                   )        136
(CUSTMAST) (L0) (SEG60FIL)    ( 6) (2) (  22) ( 10) (C)                                                                    137
** CLSUN00  TYPE 1  SEGMENT   2 LEVEL 2 CONTAINS OVERLAPPING FIELDS.                                                        *****
                   ORDRDATE OVERLAPS ORDMONTH ORDERDAY ORDRYEAR                                                            *****
                   ORDDUDAT OVERLAPS DUEMONTH DUEDAY   DUEYEAR                                                             *****
** CLSDC02  TYPE 0  THE DEFINITION IS VALID AND HAS BEEN CATALOGED.                                                        *****

FEB 11, 2002  14.27.06                                                                                                  PAGE      7
                                                  FILE STRUCTURE DIAGRAM FOR                                     DIAGRAM PAGE  1
                                                  FILE DEFINITION - CUSTMAST
RECORD FORMAT = VARIABLE BLOCKED                                                                  FILE IDENTIFICATION =
RECORD SIZE = 2492    BLOCK SIZE = 2500                                                           NUMBER OF SEGMENTS IN FILE = 6

                                                          +---------+
                                                          °CUSTOMER °
                                                          +---------+
                                                               °
                                                         +-----------+
                                                         °           °
                                                    +---------+ +---------+
                                                    °ORDER    ° °INSTALL  °
                                                    +---------+ +---------+
                                                         °
                                                   +-----------+
                                                   °           °
                                              +---------+ +---------+
                                              °SHIPINV  ° °ITEMORD  °
                                              +---------+ +---------+
                                                   °
                                                   °
                                                   °
                                              +---------+
                                              °ITEMSHIP °
                                              +---------+
FEB 11, 2002  14.27.06                                                                                                  PAGE      8
                                          DETAILED GLOSSARY BY NAME FOR                                       GLOSSARY PAGE      1
                                            FILE DEFINITION - CUSTMAST
DATE CATALOGED - 02/11/02 14:27:06                                                  DATE LAST UPDATED - 02/11/02 14:27:06
CATALOGED BY -                                                                      LAST UPDATED BY -
EXPIRATION DATE -                                                                   DATE LAST USED -
FILE IDENTIFICATION =                                                               RECORD FORMAT = VARIABLE BLOCKED
NUMBER OF SEGMENTS IN FILE = 6                                                      RECORD SIZE = 2492
NUMBER OF FIELDS IN FILE = 50                                                       BLOCK SIZE = 2500

                                        ******************************
                                        *    SEGMENT   1, LEVEL 1    *
                                        *   SEGMENT NAME = CUSTOMER  *
                                        ******************************
SEGMENT OCCURS N TIMES = 1                                          KEY FIELD 1 = CUSTNO     TYPE = C   LENGTH = 5
SEGMENT SIZE = 104
NUMBER OF FIELDS IN SEGMENT = 9
-------------------------------------------------------------------------------------------------------------------------------------
FIELD   FIELD   FIELD    FIELD   FIELD    DEC    CNT FIELD  EDIT CODES     EDIT    OUTPUT   LINE  DATA BASE NAME/
NAME    TYPE   LOCATION  LENGTH  RNDING  PLACES  FOR SGMT   ( ) ( ) ( )   LENGTH   WIDTH     NO   *** COLUMN HEADING ***
-------------------------------------------------------------------------------------------------------------------------------------
CITY      C        71       23                                                23       23
CUSTADDR  C        46       25                                                25       25
                                                                                                1    ***    CUSTOMER    ***
                                                                                                2    ***    ADDRESS     ***
CUSTNAME  C         6       30                                                30       30
                                                                                                1    ***    CUSTOMER    ***
                                                                                                2    ***      NAME      ***
CUSTNO    C         1        5                                                 5        8
                                                                                                1    ***    CUSTOMER    ***
                                                                                                2    ***     NUMBER     ***
CUSTPH    C        36       10                                                10       10
                                                                                                1    ***    CUSTOMER    ***
                                                                                                2    ***     PHONE      ***
CUSTZIP   C        96        5                                                 5        5
                                                                                                1    ***      ZIP       ***
```

Figure D-62 Definition Run: CUSTMAST File Definition (Page 4 of 7)

```
                                                                                            2   ***    CODE      ***
SEG20CNT  F       101       2                    2                      7        7
SEG60CNT  F       103       2                    6                      7        7
STATE     C        94       2                                           2        5
                                                                                                ***              ***
                                                                                            2   ***    STATE     ***
FEB 11, 2002  14.27.06                                                                                                PAGE      9
                                           DETAILED GLOSSARY BY NAME FOR                                     GLOSSARY PAGE      2
                                             FILE DEFINITION - CUSTMAST

                                          *******************************
                                          *    SEGMENT    2, LEVEL 2     *
                                          *   SEGMENT NAME = ORDER       *
                                          *******************************
COUNT FIELD FOR SEGMENT = SEG20CNT     SEGMENT ORDER =  ASCENDING     KEY FIELD 1 = ORDERNO     TYPE = C    LENGTH = 5
SEGMENT SIZE = 43
NUMBER OF FIELDS IN SEGMENT = 15
-----------------------------------------------------------------------------------------------------------------------------------
FIELD     FIELD   FIELD     FIELD   FIELD     DEC    CNT FIELD  EDIT CODES    EDIT    OUTPUT   LINE  DATA BASE NAME/
NAME      TYPE   LOCATION  LENGTH  RNDING   PLACES   FOR SGMT   ( ) ( ) ( )  LENGTH   WIDTH    NO   *** COLUMN HEADING ***
-----------------------------------------------------------------------------------------------------------------------------------
DUEDAY    C        24       2                                           2        3
                                                                                            1   ***    DAY       ***
                                                                                            2   ***    DUE       ***
DUEMONTH  C        22       2                                           2        5
                                                                                            1   ***    MONTH     ***
                                                                                            2   ***    DUE       ***
DUEYEAR   C        26       2                                           2        4
                                                                                            1   ***    YEAR      ***
                                                                                            2   ***    DUE       ***
ORDCMPLT  C        29       1                                           1       10
                                                                                            1   ***    ORDER     ***
                                                                                            2   ***  COMPLETED?  ***
ORDDUDAT  C        22       6                                           6        6
                                                                                            1   ***    DUE       ***
                                                                                            2   ***    DATE      ***
ORDERDAY  C        10       2                                           2        5
                                                                                            1   ***    ORDER     ***
                                                                                            2   ***    DAY       ***
ORDERNO   C         1       5                                           5        6
                                                                                            1   ***    ORDER     ***
                                                                                            2   ***    NUMBER    ***
ORDINVGN  C        28       1                                           1       10
                                                                                            1   ***    INVOICE   ***
                                                                                            2   ***  GENERATED?  ***
ORDMONTH  C         8       2                                           2        5
                                                                                            1   ***    ORDER     ***
                                                                                            2   ***    MONTH     ***
ORDPONUM  C        17       5                                           5       12
                                                                                            1   ***  PURCH. ORDER ***
                                                                                            2   ***    NUMBER    ***
FEB 11, 2002  14.27.06                                                                                                PAGE     10
                                           DETAILED GLOSSARY BY NAME FOR                                     GLOSSARY PAGE      3
                                             FILE DEFINITION - CUSTMAST
                                          SEGMENT    2, LEVEL 2 (CONTINUED)
-----------------------------------------------------------------------------------------------------------------------------------
FIELD     FIELD   FIELD     FIELD   FIELD     DEC    CNT FIELD  EDIT CODES    EDIT    OUTPUT   LINE  DATA BASE NAME/
NAME      TYPE   LOCATION  LENGTH  RNDING   PLACES   FOR SGMT   ( ) ( ) ( )  LENGTH   WIDTH    NO   *** COLUMN HEADING ***
-----------------------------------------------------------------------------------------------------------------------------------
ORDRDATE  C         8       6                                           6        6
                                                                                            1   ***    ORDER     ***
                                                                                            2   ***    DATE      ***
ORDRYEAR  C        12       2                                           2        5
                                                                                            1   ***    ORDER     ***
                                                                                            2   ***    YEAR      ***
ORPERSON  C        14       3                                           3        6
                                                                                            1   ***    SALES     ***
                                                                                            2   ***    PERSON    ***
SEG30CNT  F        40       2                    3                      7        7
SEG50CNT  F        42       2                    5                      7        7
FEB 11, 2002  14.27.06                                                                                                PAGE     11
                                           DETAILED GLOSSARY BY NAME FOR                                     GLOSSARY PAGE      4
                                             FILE DEFINITION - CUSTMAST

                                          *******************************
                                          *    SEGMENT    3, LEVEL 3     *
                                          *   SEGMENT NAME = SHIPINV     *
                                          *******************************
COUNT FIELD FOR SEGMENT = SEG30CNT     SEGMENT ORDER =  ASCENDING     KEY FIELD 1 = SHIPNO      TYPE = C    LENGTH = 4
SEGMENT SIZE = 100
NUMBER OF FIELDS IN SEGMENT = 11
-----------------------------------------------------------------------------------------------------------------------------------
FIELD     FIELD   FIELD     FIELD   FIELD     DEC    CNT FIELD  EDIT CODES    EDIT    OUTPUT   LINE  DATA BASE NAME/
NAME      TYPE   LOCATION  LENGTH  RNDING   PLACES   FOR SGMT   ( ) ( ) ( )  LENGTH   WIDTH    NO   *** COLUMN HEADING ***
-----------------------------------------------------------------------------------------------------------------------------------
CHECKNO   C        76       7                                           7        7
```

Figure D-62     Definition Run: CUSTMAST File Definition (Page 5 of 7)

```
                                                                                                  1   ***    CHECK     ***
                                                                                                  2   ***    NUMBER    ***
FRTCOST   P       23       4            2             ($) ( ) ( )    10      10
                                                                                                  1   ***   FREIGHT    ***
                                                                                                  2   ***    COST      ***
INAMT     P       54       4            2             ($) ( ) ( )    10      10
                                                                                                  1   ***   INVOICE    ***
                                                                                                  2   ***   AMOUNT     ***
INVDATE   C       48       6                                          6       7
                                                                                                  1   ***   INVOICE    ***
                                                                                                  2   ***    DATE      ***
INVNO     C       41       5                                          5       7
                                                                                                  1   ***   INVOICE    ***
                                                                                                  2   ***   NUMBER     ***
PAIDDATE  C       83       6                                          6       7
                                                                                                  1   ***   DATE OF    ***
                                                                                                  2   ***   PAYMENT    ***
PAYDATE   C       70       6                                          6       7
                                                                                                  1   ***   PAYMENT    ***
                                                                                                  2   ***    DATE      ***
SEG40CNT  F       99       2                    4                     7       7
SHIPDATE  C        5       6                                          6       7
                                                                                                  1   ***    DATE      ***
                                                                                                  2   ***   SHIPPED    ***
SHIPNO    C        1       4                                          4       8
                                                                                                  1   ***   SHIPPING   ***
                                                                                                  2   ***    NUMBER    ***
SHIPPER   C       11       3                                          3       7
                                                                                                  1   ***   SHIPPER    ***
FEB 11, 2002  14.27.06                                                                                                PAGE     12
                                        DETAILED GLOSSARY BY NAME FOR                                        GLOSSARY PAGE      5
                                          FILE DEFINITION - CUSTMAST

                                       ********************************
                                       *    SEGMENT   4, LEVEL 4     *
                                       *   SEGMENT NAME = ITEMSHIP   *
                                       ********************************
COUNT FIELD FOR SEGMENT = SEG40CNT     SEGMENT ORDER =  ASCENDING     KEY FIELD 1 = ITEMSHIP   TYPE = C   LENGTH = 7
SEGMENT SIZE = 24
NUMBER OF FIELDS IN SEGMENT = 3
-----------------------------------------------------------------------------------------------------------------------------
FIELD   FIELD   FIELD    FIELD   FIELD    DEC    CNT FIELD  EDIT CODES     EDIT    OUTPUT   LINE  DATA BASE NAME/
NAME    TYPE   LOCATION  LENGTH  RNDING  PLACES  FOR SGMT   ( ) ( ) ( )   LENGTH   WIDTH     NO   *** COLUMN HEADING ***
-----------------------------------------------------------------------------------------------------------------------------
ITEMSHIP  C        1       7                                          7       9
                                                                                                  1   ***   STOCK NO.    ***
                                                                                                  2   ***    SHIPPED     ***
QTYSHIP   Z        8       7                                         10      10
                                                                                                  1   ***    QUANTITY    ***
                                                                                                  2   ***    SHIPPED     ***
SEG40FIL  C       15      10                                         10      10
FEB 11, 2002  14.27.06                                                                                                PAGE     13
                                        DETAILED GLOSSARY BY NAME FOR                                        GLOSSARY PAGE      6
                                          FILE DEFINITION - CUSTMAST

                                       ********************************
                                       *    SEGMENT   5, LEVEL 3     *
                                       *   SEGMENT NAME = ITEMORD    *
                                       ********************************
COUNT FIELD FOR SEGMENT = SEG50CNT     SEGMENT ORDER =  ASCENDING     KEY FIELD 1 = ITEMORD    TYPE = C   LENGTH = 7
SEGMENT SIZE = 57
NUMBER OF FIELDS IN SEGMENT = 6
-----------------------------------------------------------------------------------------------------------------------------
FIELD   FIELD   FIELD    FIELD   FIELD    DEC    CNT FIELD  EDIT CODES     EDIT    OUTPUT   LINE  DATA BASE NAME/
NAME    TYPE   LOCATION  LENGTH  RNDING  PLACES  FOR SGMT   ( ) ( ) ( )   LENGTH   WIDTH     NO   *** COLUMN HEADING ***
-----------------------------------------------------------------------------------------------------------------------------
ITEMNAME  C        8      20                                         20      20
                                                                                                  1   ***  DESCRIPTION   ***
ITEMORD   C        1       7                                          7       9
                                                                                                  1   ***   STOCK NO.    ***
                                                                                                  2   ***    ORDERED     ***
ITMPRICE  Z       28       6            2             ($) ( ) ( )     9       9
                                                                                                  1   ***     PRICE      ***
ITMQTYOR  Z       34       7                                         10      10
                                                                                                  1   ***     ORDER      ***
                                                                                                  2   ***    QUANTITY    ***
QTYBKORD  Z       41       7                                         10      11
                                                                                                  1   ***   QUANTITY     ***
                                                                                                  2   ***  BACKORDERED   ***
SEG50FIL  C       48      10                                         10      10
FEB 11, 2002  14.27.06                                                                                                PAGE     14
                                        DETAILED GLOSSARY BY NAME FOR                                        GLOSSARY PAGE      7
                                          FILE DEFINITION - CUSTMAST

                                       ********************************
                                       *    SEGMENT   6, LEVEL 2     *
```

Figure D-62 Definition Run: CUSTMAST File Definition (Page 6 of 7)

```
                                        *   SEGMENT NAME = INSTALL    *
                                        ********************************
COUNT FIELD FOR SEGMENT = SEG60CNT     SEGMENT ORDER =  ASCENDING     KEY FIELD 1 = INSTNO     TYPE = C   LENGTH = 4
SEGMENT SIZE = 31
NUMBER OF FIELDS IN SEGMENT = 6
-------------------------------------------------------------------------------------------------------------------------------
FIELD   FIELD   FIELD    FIELD   FIELD    DEC    CNT FIELD  EDIT CODES     EDIT    OUTPUT   LINE  DATA BASE NAME/
NAME    TYPE   LOCATION  LENGTH  RNDING  PLACES  FOR SGMT   ( ) ( ) ( )   LENGTH   WIDTH    NO   *** COLUMN HEADING ***
-------------------------------------------------------------------------------------------------------------------------------
INPRODID  C        18        2                                              2        7
                                                                                      1    ***    PRODUCT     ***
                                                                                      2    ***    RELEASE     ***
INSTCONT  C         5        3                                              3       12
                                                                                      1    ***  INSTALLATION  ***
                                                                                      2    ***    CONTACT     ***
INSTNO    C         1        4                                              4       12
                                                                                      1    ***  INSTALLATION  ***
                                                                                      2    ***    NUMBER      ***
INSTOPS   C         8        8                                              8        9
                                                                                      1    ***   OPERATING    ***
                                                                                      2    ***    SYSTEM      ***
INSTOSRL  C        16        2                                              2        9
                                                                                      1    ***   OP SYSTEM    ***
                                                                                      2    ***    RELEASE     ***
SEG60FIL  C        22       10                                             10       10
** CLSGB00  TYPE 0  END OF FILE DEFINITION GLOSSARY.                                                                    *****
```

Figure D-62 Definition Run: CUSTMAST File Definition (Page 7 of 7)

## ITEMDAT1 File Definition

```
FEB 11, 2002  14.27.06                                                                                          PAGE     15
 ** CLSAA00  TYPE 0  DEFINITION ITEMDAT1 IS BEING PROCESSED.                                                             *****
-------------------------------------------------------------------------------------------------------------------------------
   FILE    STMT    FILE    DELETE GLOSSARY RECORD RECORD RECORDS BUFFER MAPPED      MAPPING         MAPPED          SEQUENCE   LIST
   NAME    TYPE    IDENT    FILE   LEVEL   FORMAT  SIZE  PER BLK  SIZE   FILE    INIT      TERM    RECSIZ         73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(ITEMDAT1) (FD)             (R)             (F)    (  80)                                                                  138
COMMENT ***** ITEM DAT1 CONTAINS DATA TO CREATE THE ITEM MAST            *****                                              139
-------------------------------------------------------------------------------------------------------------------------------
   FILE    STMT  SEGMENT  DEL SGMT VRT             SUP SEG  KEY                 AUTH          EXTERNAL          SEQUENCE   LIST
   NAME    TYPE   NAME    ETE NMBR SEG             DPL KEY CHAR                 ID          TABLE NAME          73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(ITEMDAT1) (LS) (ITEMSEG )    ( 1)                                                                                      140
-------------------------------------------------------------------------------------------------------------------------------
   FILE    STMT   FIELD    DEL SGMT LEV FIELD  FIELD FLD SEG FLD DEC CNT OCRS OUTP EDIT     DATA BASE NAME/          SEQUENCE   LIST
   NAME    TYPE   NAME    ETE NMBR NUM LOCATN LNGTH TYP KEY RND PLC SEG NTIM EDIT LNGH     COLUMN HEADING           73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(ITEMDAT1) (L0) (ITEMNO  )    ( 1) (1) (   1) (  7) (C) (1)                                                             141
(ITEMDAT1) (L1) (ITEMNO  )                                                        (ITEM                    )            142
(ITEMDAT1) (L2) (ITEMNO  )                                                        (NUMBER                  )            143
(ITEMDAT1) (L0) (ITEMNAME)    ( 1) (1) (   9) ( 20) (C)                                                                 144
(ITEMDAT1) (L1) (ITEMNAME)                                                        (ITEM                    )            145
(ITEMDAT1) (L2) (ITEMNAME)                                                        (NAME                    )            146
(ITEMDAT1) (L0) (ITMPRICE)    ( 1) (1) (  30) (  6) (Z)         (2)         ($  )                                       147
(ITEMDAT1) (L1) (ITMPRICE)                                                        (ITEM                    )            148
(ITEMDAT1) (L2) (ITMPRICE)                                                        (PRICE                   )            149
** CLSDC02  TYPE 0  THE DEFINITION IS VALID AND HAS BEEN CATALOGED.                                                      *****
FEB 11, 2002  14.27.06                                                                                          PAGE     16
                                      DETAILED GLOSSARY BY NAME FOR                                          GLOSSARY PAGE      1
                                        FILE DEFINITION - ITEMDAT1

DATE CATALOGED - 02/11/02 14:27:06                          DATE LAST UPDATED - 02/11/02 14:27:06
CATALOGED BY -                                              LAST UPDATED BY -
EXPIRATION DATE -                                           DATE LAST USED -
FILE IDENTIFICATION =                                       RECORD FORMAT = FIXED UNBLOCKED
NUMBER OF SEGMENTS IN FILE = 1                              RECORD SIZE = 80
NUMBER OF FIELDS IN FILE = 3                                BLOCK SIZE = 80

                                      ********************************
                                      *    SEGMENT   1, LEVEL 1     *
                                      *   SEGMENT NAME = ITEMSEG    *
                                      ********************************
SEGMENT OCCURS N TIMES = 1                                KEY FIELD 1 = ITEMNO     TYPE = C   LENGTH = 7
SEGMENT SIZE = 35
NUMBER OF FIELDS IN SEGMENT = 3
-------------------------------------------------------------------------------------------------------------------------------
FIELD   FIELD   FIELD    FIELD   FIELD    DEC    CNT FIELD  EDIT CODES     EDIT    OUTPUT   LINE  DATA BASE NAME/
NAME    TYPE   LOCATION  LENGTH  RNDING  PLACES  FOR SGMT   ( ) ( ) ( )   LENGTH   WIDTH    NO   *** COLUMN HEADING ***
-------------------------------------------------------------------------------------------------------------------------------
ITEMNAME  C         9       20                                             20       20
```

Figure D-63 Definition Run: ITEMDAT1 File Definition (Page 1 of 2)

```
                                                                                                                                1   ***     ITEM      ***
                                                                                                                                2   ***     NAME      ***
ITEMNO    C        1        7                                                                    7        7
                                                                                                                                1   ***     ITEM      ***
                                                                                                                                2   ***     NUMBER    ***
ITMPRICE  Z       30        6             2              ($) ( ) ( )       9        9
                                                                                                                                1   ***     ITEM      ***
                                                                                                                                2   ***     PRICE     ***
** CLSGB00  TYPE 0  END OF FILE DEFINITION GLOSSARY.                                                                                                    *****
```

Figure D-63 Definition Run: ITEMDAT1 File Definition (Page 2 of 2)

## ITEMMAST File Definition

```
FEB 11, 2002  14.27.06                                                                                                                  PAGE     17
** CLSAA00  TYPE 0  DEFINITION ITEMMAST IS BEING PROCESSED.                                                                                        *****
-------------------------------------------------------------------------------------------------------------------------------------------------------
   FILE     STMT    FILE    DELETE GLOSSARY RECORD RECORD RECORDS BUFFER MAPPED      MAPPING         MAPPED           SEQUENCE   LIST
   NAME     TYPE    IDENT    FILE   LEVEL   FORMAT  SIZE  PER BLK  SIZE   FILE    INIT      TERM     RECSIZ          73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------------------------------
(ITEMMAST) (FD)             (R)             (K)                 (   33)                                                                    150
COMMENT ***** ITEM FILE CONTAINS ITEM NAME AND WEIGHT                          *****                                                          151
-------------------------------------------------------------------------------------------------------------------------------------------------------
   FILE     STMT  SEGMENT  DEL SGMT VRT             SUP SEG  KEY                AUTH          EXTERNAL             SEQUENCE   LIST
   NAME     TYPE   NAME    ETE NMBR SEG             DPL KEY CHAR                ID          TABLE NAME             73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------------------------------
(ITEMMAST) (LS) (ITEMSEG )    ( 1)                                                                                                          152
-------------------------------------------------------------------------------------------------------------------------------------------------------
   FILE     STMT   FIELD    DEL SGMT LEV FIELD  FIELD FLD SEG FLD DEC CNT OCRS OUTP EDIT      DATA BASE NAME/          SEQUENCE   LIST
   NAME     TYPE   NAME     ETE NMBR NUM LOCATN LNGTH TYP KEY RND PLC SEG NTIM EDIT LNGH      COLUMN HEADING           73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------------------------------
(ITEMMAST) (L0) (ITEMNO  )    ( 1) (1) (   1) (  7) (C) (1)                                                                                 153
(ITEMMAST) (L1) (ITEMNO  )                                                                 (ITEM                    )                   154
(ITEMMAST) (L2) (ITEMNO  )                                                                 (NUMBER                  )                   155
(ITEMMAST) (L0) (ITEMNAME)    ( 1) (1) (   8) ( 20) (C)                                                                                     156
(ITEMMAST) (L1) (ITEMNAME)                                                                 (ITEM                    )                   157
(ITEMMAST) (L2) (ITEMNAME)                                                                 (NAME                    )                   158
(ITEMMAST) (L0) (ITMPRICE)    ( 1) (1) (  28) (  6) (Z)         (2)          ($  )                                                        159
(ITEMMAST) (L1) (ITMPRICE)                                                                 (ITEM                    )                   160
(ITEMMAST) (L2) (ITMPRICE)                                                                 (PRICE                   )                   161
** CLSDC02  TYPE 0  THE DEFINITION IS VALID AND HAS BEEN CATALOGED.                                                                             *****
FEB 11, 2002  14.27.06                                                                                                                  PAGE     18
                                        DETAILED GLOSSARY BY NAME FOR                                                          GLOSSARY PAGE      1
                                          FILE DEFINITION - ITEMMAST

DATE CATALOGED - 02/11/02 14:27:06                                    DATE LAST UPDATED - 02/11/02 14:27:06
CATALOGED BY -                                                        LAST UPDATED BY -
EXPIRATION DATE -                                                     DATE LAST USED -
FILE IDENTIFICATION =                                                 RECORD FORMAT = KEY SEQUENCED VSAM
NUMBER OF SEGMENTS IN FILE = 1                                        RECORD SIZE = 33
NUMBER OF FIELDS IN FILE = 3                                          BUFFER SIZE = 33

                                      ********************************
                                      *    SEGMENT   1, LEVEL 1      *
                                      *   SEGMENT NAME = ITEMSEG     *
                                      ********************************
SEGMENT OCCURS N TIMES = 1                                  KEY FIELD 1 = ITEMNO     TYPE = C   LENGTH = 7
SEGMENT SIZE = 33
NUMBER OF FIELDS IN SEGMENT = 3
SEGMENT KEY CHARACTERISTICS = U
-------------------------------------------------------------------------------------------------------------------------------------------------------
FIELD   FIELD   FIELD    FIELD   FIELD    DEC    CNT FIELD  EDIT CODES     EDIT    OUTPUT   LINE  DATA BASE NAME/
NAME    TYPE   LOCATION  LENGTH  RNDING  PLACES  FOR SGMT   ( ) ( ) ( )   LENGTH  WIDTH     NO   *** COLUMN HEADING ***
-------------------------------------------------------------------------------------------------------------------------------------------------------
ITEMNAME  C        8       20                                                 20       20
                                                                                                1   ***     ITEM      ***
                                                                                                2   ***     NAME      ***
ITEMNO    C        1        7                                                  7        7
                                                                                                1   ***     ITEM      ***
                                                                                                2   ***     NUMBER    ***
ITMPRICE  Z       28        6             2              ($) ( ) ( )       9        9
                                                                                                1   ***     ITEM      ***
                                                                                                2   ***     PRICE     ***
** CLSGB00  TYPE 0  END OF FILE DEFINITION GLOSSARY.                                                                                            *****
```

Figure D-64 Definition Run: ITEMMAST File Definition

## TRANWORK File Definition

```
FEB 11, 2002  14.27.06                                                                                                PAGE     19
** CLSAA00  TYPE 0  DEFINITION TRANWORK IS BEING PROCESSED.                                                                *****

   FILE    STMT    FILE    DELETE GLOSSARY RECORD RECORD RECORDS BUFFER MAPPED      MAPPING          MAPPED          SEQUENCE   LIST
   NAME    TYPE    IDENT    FILE   LEVEL   FORMAT  SIZE  PER BLK  SIZE   FILE    INIT      TERM    RECSIZ          73 TO 80   SEQ
(TRANWORK) (FD)             (R)             (F)    (  80)  ( 10)                                                          162

   FILE    STMT  SEGMENT  DEL SGMT VRT            SUP SEG  KEY              AUTH          EXTERNAL          SEQUENCE   LIST
   NAME    TYPE   NAME    ETE NMBR SEG            DPL KEY CHAR              ID          TABLE NAME          73 TO 80   SEQ
(TRANWORK) (LS) (WORK01  )    ( 1)                                                                                      163

   FILE    STMT    FIELD   DEL SGMT LEV FIELD  FIELD FLD SEG FLD DEC CNT OCRS OUTP EDIT      DATA BASE NAME/          SEQUENCE   LIST
   NAME    TYPE    NAME    ETE NMBR NUM LOCATN LNGTH TYP KEY RND PLC SEG NTIM EDIT LNGH      COLUMN HEADING           73 TO 80   SEQ
(TRANWORK) (L0) (TRANID  )    ( 1) (1) (   1) (  8) (C)                                                                 164
(TRANWORK) (L1) (TRANID  )                                                          (TRANS                 )            165
(TRANWORK) (L2) (TRANID  )                                                          (ID                    )            166
(TRANWORK) (L0) (CUSTNO  )    ( 1) (1) (   9) (  5) (C) (1)                                                             167
(TRANWORK) (L1) (CUSTNO  )                                                          (CUST                  )            168
(TRANWORK) (L2) (CUSTNO  )                                                          (NO.                   )            169
(TRANWORK) (L0) (ORDERNO )    ( 1) (1) (  14) (  5) (C)                                                                 170
(TRANWORK) (L1) (ORDERNO )                                                          (ORDER                 )            171
(TRANWORK) (L2) (ORDERNO )                                                          (NO.                   )            172
(TRANWORK) (L0) (ORDRDATE)    ( 1) (1) (  19) (  6) (C)                                                                 173
(TRANWORK) (L1) (ORDRDATE)                                                          (ORDER                 )            174
(TRANWORK) (L2) (ORDRDATE)                                                          (DATE                  )            175
(TRANWORK) (L0) (ORDRPERS)    ( 1) (1) (  25) (  3) (C)                                                                 176
(TRANWORK) (L1) (ORDRPERS)                                                          (ORDER                 )            177
(TRANWORK) (L2) (ORDRPERS)                                                          (PERSON                )            178
(TRANWORK) (L0) (ORDRPONO)    ( 1) (1) (  28) (  5) (C)                                                                 179
(TRANWORK) (L1) (ORDRPONO)                                                          (ORDER                 )            180
(TRANWORK) (L2) (ORDRPONO)                                                          (PO                    )            181
(TRANWORK) (L0) (ORDRDUED)    ( 1) (1) (  33) (  6) (C)                                                                 182
(TRANWORK) (L1) (ORDRDUED)                                                          (DUE                   )            183
(TRANWORK) (L2) (ORDRDUED)                                                          (DATE                  )            184
(TRANWORK) (L0) (ORDRINVG)    ( 1) (1) (  39) (  1) (C)                                                                 185
(TRANWORK) (L1) (ORDRINVG)                                                          (INV                   )            186
(TRANWORK) (L2) (ORDRINVG)                                                          (GEN                   )            187
(TRANWORK) (L0) (ORDRCOMP)    ( 1) (1) (  40) (  1) (C)                                                                 188
(TRANWORK) (L1) (ORDRCOMP)                                                          (INV                   )            189
(TRANWORK) (L2) (ORDRCOMP)                                                          (CMP                   )            190
(TRANWORK) (L0) (ITM5ORD1)    ( 1) (1) (  19) (  7) (C)                                                                 191
(TRANWORK) (L1) (ITM5ORD1)                                                          (ITEM                  )            192
(TRANWORK) (L2) (ITM5ORD1)                                                          (ORDR 1                )            193
(TRANWORK) (L0) (ITM5QTY1)    ( 1) (1) (  26) (  5) (Z)                                                                 194
(TRANWORK) (L1) (ITM5QTY1)                                                          (ITEM                  )            195
(TRANWORK) (L2) (ITM5QTY1)                                                          (QTY 1                 )            196
(TRANWORK) (L0) (ITM5ORD2)    ( 1) (1) (  31) (  7) (C)                                                                 197
(TRANWORK) (L1) (ITM5ORD2)                                                          (ITEM                  )            198
(TRANWORK) (L2) (ITM5ORD2)                                                          (ORDR 2                )            199
(TRANWORK) (L0) (ITM5QTY2)    ( 1) (1) (  38) (  5) (Z)                                                                 200
(TRANWORK) (L1) (ITM5QTY2)                                                          (ITEM                  )            201
(TRANWORK) (L2) (ITM5QTY2)                                                          (QTY 2                 )            202
(TRANWORK) (L0) (ITM5ORD3)    ( 1) (1) (  43) (  7) (C)                                                                 203
(TRANWORK) (L1) (ITM5ORD3)                                                          (ITEM                  )            204
(TRANWORK) (L2) (ITM5ORD3)                                                          (ORDR 3                )            205
(TRANWORK) (L0) (ITM5QTY3)    ( 1) (1) (  50) (  5) (Z)                                                                 206
FEB 11, 2002  14.27.06                                                                                                PAGE     20

   FILE    STMT    FIELD   DEL SGMT LEV FIELD  FIELD FLD SEG FLD DEC CNT OCRS OUTP EDIT      DATA BASE NAME/          SEQUENCE   LIST
   NAME    TYPE    NAME    ETE NMBR NUM LOCATN LNGTH TYP KEY RND PLC SEG NTIM EDIT LNGH      COLUMN HEADING           73 TO 80   SEQ
(TRANWORK) (L1) (ITM5QTY3)                                                          (ITEM                  )            207
(TRANWORK) (L2) (ITM5QTY3)                                                          (QTY 3                 )            208
(TRANWORK) (L0) (ITM5ORD4)    ( 1) (1) (  55) (  7) (C)                                                                 209
(TRANWORK) (L1) (ITM5ORD4)                                                          (ITEM                  )            210
(TRANWORK) (L2) (ITM5ORD4)                                                          (ORDR 4                )            211
(TRANWORK) (L0) (ITM5QTY4)    ( 1) (1) (  62) (  5) (Z)                                                                 212
(TRANWORK) (L1) (ITM5QTY4)                                                          (ITEM                  )            213
(TRANWORK) (L2) (ITM5QTY4)                                                          (QTY 4                 )            214
(TRANWORK) (L0) (ITM5ORD5)    ( 1) (1) (  67) (  7) (C)                                                                 215
(TRANWORK) (L1) (ITM5ORD5)                                                          (ITEM                  )            216
(TRANWORK) (L2) (ITM5ORD5)                                                          (ORDR 5                )            217
(TRANWORK) (L0) (ITM5QTY5)    ( 1) (1) (  74) (  5) (Z)                                                                 218
(TRANWORK) (L1) (ITM5QTY5)                                                          (ITEM                  )            219
(TRANWORK) (L2) (ITM5QTY5)                                                          (QTY 5                 )            220
(TRANWORK) (L0) (ALL80   )    ( 1) (1) (   1) ( 80) (C)                                                                 221
(TRANWORK) (L1) (ALL80   )                                                          (TRAN RECORD           )            222
(TRANWORK) (L2) (ALL80   )                                                          (ALL 80 POSITIONS      )            223
** CLSUN00  TYPE 1  SEGMENT   1 LEVEL 1 CONTAINS OVERLAPPING FIELDS.                                                     *****
```

Figure D-65 Definition Run: TRANWORK File Definition (Page 1 of 3)

```
                    ALL80    OVERLAPS TRANID   CUSTNO   ORDERNO  ITM5ORD1                                        *****
                                      ORDRDATE ORDRPERS ITM5QTY1 ORDRPONO                                        *****
                                      ITM5ORD2 ORDRDUED ITM5QTY2 ORDRINVG                                        *****
                                      ORDRCOMP ITM5ORD3 ITM5QTY3 ITM5ORD4                                        *****
                                      ITM5QTY4 ITM5ORD5 ITM5QTY5                                                 *****
                    ITM5ORD1 OVERLAPS ORDRDATE ORDRPERS                                                          *****
                    ORDRPERS OVERLAPS ITM5QTY1                                                                   *****
                    ITM5QTY1 OVERLAPS ORDRPONO                                                                   *****
                    ORDRPONO OVERLAPS ITM5ORD2                                                                   *****
                    ITM5ORD2 OVERLAPS ORDRDUED                                                                   *****
                    ORDRDUED OVERLAPS ITM5QTY2                                                                   *****
                    ITM5QTY2 OVERLAPS ORDRINVG ORDRCOMP                                                          *****
** CLSDC02  TYPE 0  THE DEFINITION IS VALID AND HAS BEEN CATALOGED.                                             *****
FEB 11, 2002  14.27.06                                                                              PAGE     21
                                        DETAILED GLOSSARY BY NAME FOR                          GLOSSARY PAGE      1
                                          FILE DEFINITION - TRANWORK

DATE CATALOGED - 02/11/02 14:27:06                      DATE LAST UPDATED - 02/11/02 14:27:06
CATALOGED BY -                                          LAST UPDATED BY -
EXPIRATION DATE -                                       DATE LAST USED -
FILE IDENTIFICATION =                                   RECORD FORMAT = FIXED BLOCKED
NUMBER OF SEGMENTS IN FILE = 1                          RECORD SIZE = 80
NUMBER OF FIELDS IN FILE = 20                           BLOCK SIZE = 800

                                      ********************************
                                      *    SEGMENT   1, LEVEL 1     *
                                      *   SEGMENT NAME = WORK01     *
                                      ********************************

SEGMENT OCCURS N TIMES = 1                                KEY FIELD 1 = CUSTNO     TYPE = C   LENGTH = 5
SEGMENT SIZE = 80
NUMBER OF FIELDS IN SEGMENT = 20
-------------------------------------------------------------------------------------------------------------------
FIELD   FIELD   FIELD    FIELD   FIELD    DEC    CNT FIELD  EDIT CODES    EDIT    OUTPUT   LINE  DATA BASE NAME/
NAME    TYPE   LOCATION  LENGTH  RNDING  PLACES  FOR SGMT   ( ) ( ) ( )  LENGTH   WIDTH     NO  *** COLUMN HEADING ***
-------------------------------------------------------------------------------------------------------------------
ALL80     C        1       80                                              80       80
                                                                                      1   ***  TRAN RECORD   ***
                                                                                      2   ***ALL 80 POSITIONS***
CUSTNO    C        9        5                                               5        5
                                                                                      1   ***      CUST      ***
                                                                                      2   ***      NO.       ***
ITM5ORD1  C       19        7                                               7        7
                                                                                      1   ***      ITEM      ***
                                                                                      2   ***     ORDR 1     ***
ITM5ORD2  C       31        7                                               7        7
                                                                                      1   ***      ITEM      ***
                                                                                      2   ***     ORDR 2     ***
ITM5ORD3  C       43        7                                               7        7
                                                                                      1   ***      ITEM      ***
                                                                                      2   ***     ORDR 3     ***
ITM5ORD4  C       55        7                                               7        7
                                                                                      1   ***      ITEM      ***
                                                                                      2   ***     ORDR 4     ***
ITM5ORD5  C       67        7                                               7        7
                                                                                      1   ***      ITEM      ***
                                                                                      2   ***     ORDR 5     ***
ITM5QTY1  Z       26        5                                               7        7
                                                                                      1   ***      ITEM      ***
                                                                                      2   ***     QTY 1      ***
FEB 11, 2002  14.27.06                                                                              PAGE     22
                                        DETAILED GLOSSARY BY NAME FOR                          GLOSSARY PAGE      2
                                          FILE DEFINITION - TRANWORK
                                       SEGMENT   1, LEVEL 1 (CONTINUED)
-------------------------------------------------------------------------------------------------------------------
FIELD   FIELD   FIELD    FIELD   FIELD    DEC    CNT FIELD  EDIT CODES    EDIT    OUTPUT   LINE  DATA BASE NAME/
NAME    TYPE   LOCATION  LENGTH  RNDING  PLACES  FOR SGMT   ( ) ( ) ( )  LENGTH   WIDTH     NO  *** COLUMN HEADING ***
-------------------------------------------------------------------------------------------------------------------
ITM5QTY2  Z       38        5                                               7        7
                                                                                      1   ***      ITEM      ***
                                                                                      2   ***     QTY 2      ***
ITM5QTY3  Z       50        5                                               7        7
                                                                                      1   ***      ITEM      ***
                                                                                      2   ***     QTY 3      ***
ITM5QTY4  Z       62        5                                               7        7
                                                                                      1   ***      ITEM      ***
                                                                                      2   ***     QTY 4      ***
ITM5QTY5  Z       74        5                                               7        7
                                                                                      1   ***      ITEM      ***
                                                                                      2   ***     QTY 5      ***
ORDERNO   C       14        5                                               5        5
                                                                                      1   ***     ORDER      ***
                                                                                      2   ***      NO.       ***
ORDRCOMP  C       40        1                                               1        3
                                                                                      1   ***       INV      ***
                                                                                      2   ***       CMP      ***
ORDRDATE  C       19        6                                               6        6
                                                                                      1   ***     ORDER      ***
```

Figure D-65 Definition Run: TRANWORK File Definition (Page 2 of 3)

```
                                                                                                                2   ***    DATE       ***
ORDRDUED  C       33       6                                                                    6       6
                                                                                                                1   ***     DUE       ***
                                                                                                                2   ***    DATE       ***
ORDRINVG  C       39       1                                                                    1       3
                                                                                                                1   ***     INV       ***
                                                                                                                2   ***    GEN        ***
ORDRPERS  C       25       3                                                                    3       6
                                                                                                                1   ***    ORDER      ***
                                                                                                                2   ***    PERSON     ***
ORDRPONO  C       28       5                                                                    5       5
                                                                                                                1   ***    ORDER      ***
                                                                                                                2   ***     PO        ***
TRANID    C        1       8                                                                    8       8
                                                                                                                1   ***    TRANS      ***
                                                                                                                2   ***     ID        ***
** CLSGB00  TYPE 0  END OF FILE DEFINITION GLOSSARY.                                                                                    *****
```

Figure D-65 Definition Run: TRANWORK File Definition (Page 3 of 3)

## CUSTADDS Transaction Group Definition

```
FEB 11, 2002  14.27.06                                                                                                          PAGE     23
 ** CLSAA00  TYPE 0  DEFINITION CUSTADDS IS BEING PROCESSED.                                                                            *****
-------------------------------------------------------------------------------------------------------------------------------------------
  GROUP    STMT IDENT1 IDENT1 ID1 IDENT2 IDENT2 ID2 DEL GLS    FILE    DEFLT DEFLT REC RECORD RECORDS BUFFER          SEQUENCE   LIST
  NAME     TYPE  CODE  LOCATN LNG  CODE  LOCATN LNG ETE ARY    NAME    CREAT INSRT FRM  SIZE  PER BLK  SIZE          73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------------------
(CUSTADDS) (TD) (1ADD) (   1) (4) (CUST) (   5) (4) (R)     (CUSTMAST)           (F) (  80)                                        224
COMMENT ***** TRANSACTION DEFINITION FOR "CUSTMAST" FILE                  *****                                                      225
COMMENT *****                                                             *****                                                      226
COMMENT ***** ADD CUSTMAST TRANSACTION 1                                  *****                                                      227
-------------------------------------------------------------------------------------------------------------------------------------------
  GROUP    STMT IDENT1 IDENT2 DEL FIELD  FIELD FLD   FIELD    ACT DEC            EDIT PATTERN          VAL        SEQUENCE   LIST
  NAME     TYPE  CODE   CODE  ETE LOCATN LNGTH TYP   NAME     ION PLCS  MINIMUM VALUE     MAXIMUM VALUE  TYP        73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------------------
(CUSTADDS) (TL) (1ADD) (CUST)      (   9) (  5) (C) (CUSTNO  ) (C)        (Y9999                     ) (P)                          228
(CUSTADDS) (TL) (1ADD) (CUST)      (  14) ( 30) (C) (CUSTNAME) (R)                                                                  229
(CUSTADDS) (TL) (1ADD) (CUST)      (  44) ( 10) (C) (CUSTPH  ) (R)                                                                  230
(CUSTADDS) (TL) (1ADD) (CUST)      (  54) ( 25) (C) (CUSTADDR) (R)                                                                  231
-------------------------------------------------------------------------------------------------------------------------------------------
  GROUP    STMT IDENT1 IDENT1 ID1 IDENT2 IDENT2 ID2 DEL GLS    FILE    DEFLT DEFLT REC RECORD RECORDS BUFFER          SEQUENCE   LIST
  NAME     TYPE  CODE  LOCATN LNG  CODE  LOCATN LNG ETE ARY    NAME    CREAT INSRT FRM  SIZE  PER BLK  SIZE          73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------------------
(CUSTADDS) (TD) (2ADD) (   1) (4) (CUST) (   5) (4) (R)     (CUSTMAST)           (F) (  80)                                        232
COMMENT ***** ADD CUSTMAST TRANSACTION 2                                  *****                                                      233
-------------------------------------------------------------------------------------------------------------------------------------------
  GROUP    STMT IDENT1 IDENT2 DEL FIELD  FIELD FLD   FIELD    ACT DEC            EDIT PATTERN          VAL        SEQUENCE   LIST
  NAME     TYPE  CODE   CODE  ETE LOCATN LNGTH TYP   NAME     ION PLCS  MINIMUM VALUE     MAXIMUM VALUE  TYP        73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------------------
(CUSTADDS) (TL) (2ADD) (CUST)      (   9) (  5) (C) (CUSTNO  ) (M)        (Y9999                     ) (P)                          234
(CUSTADDS) (TL) (2ADD) (CUST)      (  14) ( 23) (C) (CITY    ) (R)                                                                  235
(CUSTADDS) (TL) (2ADD) (CUST)      (  37) (  2) (C) (STATE   ) (R)                                                                  236
(CUSTADDS) (TL) (2ADD) (CUST)      (  39) (  5) (C) (CUSTZIP ) (R)                                                                  237
** CLSDC02  TYPE 0  THE DEFINITION IS VALID AND HAS BEEN CATALOGED.                                                                    *****
FEB 11, 2002  14.27.06                                                                                                          PAGE     24
                                        GLOSSARY BY LEVEL NUMBER FOR                                                   GLOSSARY PAGE      1
                                        TRANSACTION GROUP - CUSTADDS

DATE CATALOGED - 02/11/02 14:27:06                                    DATE LAST UPDATED - 02/11/02 14:27:06
CATALOGED BY -                                                        LAST UPDATED BY -
EXPIRATION DATE -                                                     DATE LAST USED -
MASTER FILE NAME = CUSTMAST                                           RECORD FORMAT = FIXED UNBLOCKED
NUMBER OF IDENTIFIERS IN GROUP = 2                                    RECORD SIZE = 80
NUMBER OF ACTIONS IN GROUP = 8                                        BLOCK SIZE = 80

                                      ************************************
                                      *  1 OF 2 IDENTIFIERS IN GROUP     *
                                      ************************************
**********    ACTIONS IN IDENTIFIER = 4          IDENTIFIER ONE = (1ADD)          IDENTIFIER TWO = (CUST)
*1ADDCUST*    DEFAULT CREATE = NO                IDENTIFIER LOCATION = 1          IDENTIFIER LOCATION = 5
**********    DEFAULT INSERT = NO                IDENTIFIER LENGTH = 4            IDENTIFIER LENGTH = 4
                                                 IDENTIFIED IF EQUAL              IDENTIFIED IF EQUAL
-------------------------------------------------------------------------------------------------------------------------------------------
MASTER  .....TRANSACTION DATA.....
FILE    FIELD     FIELD     FIELD    DECIMAL   ACTION   LEVEL          MINIMUM                  MAXIMUM
FIELD   TYPE      LOCATION  LENGTH   PLACES    CODE     NUMBER         VALUE                    VALUE
-------------------------------------------------------------------------------------------------------------------------------------------
CUSTNO    C           9        5                C        1          EDIT = Y9999
CUSTNAME  C          14       30                R        1
CUSTPH    C          44       10                R        1
CUSTADDR  C          54       25                R        1
FEB 11, 2002  14.27.06                                                                                                          PAGE     25
```

Figure D-66 Definition Run: CUSTADDS Transaction Group Definition (Page 1 of 2)

```
                                        GLOSSARY BY LEVEL NUMBER FOR                                        GLOSSARY PAGE      2
                                        TRANSACTION GROUP - CUSTADDS

                                      ************************************
                                      *  2 OF 2  IDENTIFIERS IN GROUP    *
                                      ************************************
**********    ACTIONS IN IDENTIFIER = 4          IDENTIFIER ONE = (2ADD)          IDENTIFIER TWO = (CUST)
*2ADDCUST*    DEFAULT CREATE = NO                IDENTIFIER LOCATION = 1          IDENTIFIER LOCATION = 5
**********    DEFAULT INSERT = NO                IDENTIFIER LENGTH = 4            IDENTIFIER LENGTH = 4
                                                 IDENTIFIED IF EQUAL              IDENTIFIED IF EQUAL
-------------------------------------------------------------------------------------------------------------------
MASTER  .........TRANSACTION DATA.........
FILE    FIELD     FIELD     FIELD    DECIMAL   ACTION   LEVEL          MINIMUM                  MAXIMUM
FIELD   TYPE      LOCATION  LENGTH   PLACES    CODE     NUMBER         VALUE                    VALUE
-------------------------------------------------------------------------------------------------------------------
CUSTNO    C          9        5                 M        1          EDIT = Y9999
CITY      C         14       23                 R        1
STATE     C         37        2                 R        1
CUSTZIP   C         39        5                 R        1
** CLSGT00  TYPE 0  END OF TRANSACTION DEFINITION GLOSSARY.                                                    *****
```

Figure D-66 Definition Run: CUSTADDS Transaction Group Definition (Page 2 of 2)

## ORDENTRY Transaction Group Definition

```
FEB 11, 2002  14.27.06                                                                                    PAGE     26
** CLSAA00  TYPE 0  DEFINITION ORDENTRY IS BEING PROCESSED.                                                *****
-------------------------------------------------------------------------------------------------------------------
  GROUP    STMT IDENT1 IDENT1 ID1 IDENT2 IDENT2 ID2 DEL GLS    FILE    DEFLT DEFLT REC RECORD RECORDS BUFFER        SEQUENCE   LIST
  NAME     TYPE  CODE  LOCATN LNG  CODE  LOCATN LNG ETE ARY    NAME    CREAT INSRT FRM  SIZE  PER BLK  SIZE        73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------
(ORDENTRY) (TD) (3ADD) (   1) (4) (ORDR) (   5) (4) (R)     (CUSTMAST)          (F) (  80)  ( 10)                          238
COMMENT ***** ADD ORDER TRANSACTION                                          *****                                       239
-------------------------------------------------------------------------------------------------------------------
  GROUP    STMT IDENT1 IDENT2 DEL FIELD  FIELD FLD   FIELD    ACT DEC           EDIT PATTERN          VAL       SEQUENCE   LIST
  NAME     TYPE  CODE   CODE  ETE LOCATN LNGTH TYP   NAME     ION PLCS  MINIMUM VALUE     MAXIMUM VALUE  TYP       73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------
(ORDENTRY) (TL) (3ADD) (ORDR)     (   9) (  5) (C) (CUSTNO  ) (M)       (Y9999                       ) (P)                 240
(ORDENTRY) (TL) (3ADD) (ORDR)     (  14) (  5) (C) (ORDERNO ) (I)                                                          241
(ORDENTRY) (TL) (3ADD) (ORDR)     (  19) (  6) (C) (ORDRDATE) (R)                                        (D)               242
(ORDENTRY) (TL) (3ADD) (ORDR)     (  25) (  3) (C) (ORPERSON) (R)       (-9-9-9                      ) (P)                 243
(ORDENTRY) (TL) (3ADD) (ORDR)     (  28) (  5) (C) (ORDPONUM) (R)                                                          244
(ORDENTRY) (TL) (3ADD) (ORDR)     (  33) (  6) (C) (ORDDUDAT) (R)                                        (D)               245
(ORDENTRY) (TL) (3ADD) (ORDR)     (  39) (  1) (C) (ORDINVGN) (R)       (Z                           ) (P)                 246
(ORDENTRY) (TL) (3ADD) (ORDR)     (  40) (  1) (C) (ORDCMPLT) (R)                                                          247
-------------------------------------------------------------------------------------------------------------------
  GROUP    STMT IDENT1 IDENT1 ID1 IDENT2 IDENT2 ID2 DEL GLS    FILE    DEFLT DEFLT REC RECORD RECORDS BUFFER        SEQUENCE   LIST
  NAME     TYPE  CODE  LOCATN LNG  CODE  LOCATN LNG ETE ARY    NAME    CREAT INSRT FRM  SIZE  PER BLK  SIZE        73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------
(ORDENTRY) (TD) (4ADD) (   1) (4) (ITEM) (   5) (4) (R)     (CUSTMAST)      (3)  (F) (  80)  ( 10)                      248
COMMENT ***** ADD ITEM TO ORDER TRANSACTION - SINGLE ITEM PER TRANS          *****                                       249
-------------------------------------------------------------------------------------------------------------------
  GROUP    STMT IDENT1 IDENT2 DEL FIELD  FIELD FLD   FIELD    ACT DEC           EDIT PATTERN          VAL       SEQUENCE   LIST
  NAME     TYPE  CODE   CODE  ETE LOCATN LNGTH TYP   NAME     ION PLCS  MINIMUM VALUE     MAXIMUM VALUE  TYP       73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------
(ORDENTRY) (TL) (4ADD) (ITEM)     (   9) (  5) (C) (CUSTNO  ) (M)       (Y9999                       ) (P)                 250
(ORDENTRY) (TL) (4ADD) (ITEM)     (  14) (  5) (C) (ORDERNO ) (M)                                                          251
(ORDENTRY) (TL) (4ADD) (ITEM)     (  19) (  7) (C) (ITEMORD ) (M)                                                          252
(ORDENTRY) (TL) (4ADD) (ITEM)     (  26) ( 20) (C) (ITEMNAME) (R)                                                          253
(ORDENTRY) (TL) (4ADD) (ITEM)     (  47) (  6) (Z) (ITMPRICE) (R) ( 2)  (0.00          300.0          )                   254
(ORDENTRY) (TL) (4ADD) (ITEM)     (  60) (  7) (Z) (ITMQTYOR) (R)                                                          255
-------------------------------------------------------------------------------------------------------------------
  GROUP    STMT IDENT1 IDENT1 ID1 IDENT2 IDENT2 ID2 DEL GLS    FILE    DEFLT DEFLT REC RECORD RECORDS BUFFER        SEQUENCE   LIST
  NAME     TYPE  CODE  LOCATN LNG  CODE  LOCATN LNG ETE ARY    NAME    CREAT INSRT FRM  SIZE  PER BLK  SIZE        73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------
(ORDENTRY) (TD) (5ADD) (   1) (4) (ITEM) (   5) (4) (R)     (CUSTMAST)      (3)  (F) (  80)  ( 10)                      256
COMMENT ***** ADD ITEM TO ORDER TRANSACTION - FIVE ITEMS PER TRANS           *****                                       257
-------------------------------------------------------------------------------------------------------------------
  GROUP    STMT IDENT1 IDENT2 DEL FIELD  FIELD FLD   FIELD    ACT DEC           EDIT PATTERN          VAL       SEQUENCE   LIST
  NAME     TYPE  CODE   CODE  ETE LOCATN LNGTH TYP   NAME     ION PLCS  MINIMUM VALUE     MAXIMUM VALUE  TYP       73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------
(ORDENTRY) (TL) (5ADD) (ITEM)     (   9) (  5) (C) (CUSTNO  ) (M)       (Y9999                       ) (P)                 258
(ORDENTRY) (TL) (5ADD) (ITEM)     (  14) (  5) (C) (ORDERNO ) (M)                                                          259
(ORDENTRY) (TL) (5ADD) (ITEM)     (  19) (  7) (C) (ITEMORD ) (M)                                                          260
(ORDENTRY) (TL) (5ADD) (ITEM)     (  26) (  5) (Z) (ITMQTYOR) (R)                                                          261
-------------------------------------------------------------------------------------------------------------------
  GROUP    STMT IDENT1 IDENT1 ID1 IDENT2 IDENT2 ID2 DEL GLS    FILE    DEFLT DEFLT REC RECORD RECORDS BUFFER        SEQUENCE   LIST
  NAME     TYPE  CODE  LOCATN LNG  CODE  LOCATN LNG ETE ARY    NAME    CREAT INSRT FRM  SIZE  PER BLK  SIZE        73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------
(ORDENTRY) (TD) (1DEL) (   1) (4) (ORDR) (   5) (4) (R)     (CUSTMAST)          (F) (  80)  ( 10)                          262
COMMENT ***** DEL ORDER TRANSACTION                                          *****                                       263
FEB 11, 2002  14.27.06                                                                                    PAGE     27
```

Figure D-67 Definition Run: ORDENTRY Transaction Group Definition (Page 1 of 3)

```
-------------------------------------------------------------------------------------------------------------------------------
  GROUP    STMT IDENT1 IDENT2 DEL FIELD  FIELD FLD   FIELD    ACT DEC            EDIT PATTERN          VAL       SEQUENCE   LIST
  NAME     TYPE  CODE   CODE  ETE LOCATN LNGTH TYP   NAME     ION PLCS   MINIMUM VALUE     MAXIMUM VALUE  TYP       73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(ORDENTRY) (TL) (1DEL) (ORDR)      (   9) (  5) (C) (CUSTNO  ) (M)        (Y9999                      ) (P)                 264
(ORDENTRY) (TL) (1DEL) (ORDR)      (  14) (  5) (C) (ORDERNO ) (E)                                                          265
-------------------------------------------------------------------------------------------------------------------------------
  GROUP    STMT IDENT1 IDENT1 ID1 IDENT2 IDENT2 ID2 DEL GLS    FILE    DEFLT DEFLT REC RECORD RECORDS BUFFER          SEQUENCE   LIST
  NAME     TYPE  CODE  LOCATN LNG  CODE  LOCATN LNG ETE ARY    NAME    CREAT INSRT FRM  SIZE  PER BLK  SIZE          73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(ORDENTRY) (TD) (2DEL) (   1) (4) (ITEM) (   5) (4) (R)      (CUSTMAST)           (F) (  80)  ( 10)                        266
COMMENT ***** DEL ITEM FROM ORDER TRANSACTION                                  *****                                          267
-------------------------------------------------------------------------------------------------------------------------------
  GROUP    STMT IDENT1 IDENT2 DEL FIELD  FIELD FLD   FIELD    ACT DEC            EDIT PATTERN          VAL       SEQUENCE   LIST
  NAME     TYPE  CODE   CODE  ETE LOCATN LNGTH TYP   NAME     ION PLCS   MINIMUM VALUE     MAXIMUM VALUE  TYP       73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(ORDENTRY) (TL) (2DEL) (ITEM)      (   9) (  5) (C) (CUSTNO  ) (M)        (Y9999                      ) (P)                 268
(ORDENTRY) (TL) (2DEL) (ITEM)      (  14) (  5) (C) (ORDERNO ) (M)                                                          269
(ORDENTRY) (TL) (2DEL) (ITEM)      (  19) (  7) (C) (ITEMORD ) (E)                                                          270
** CLSDC02  TYPE 0  THE DEFINITION IS VALID AND HAS BEEN CATALOGED.                                                        *****
FEB 11, 2002  14.27.06                                                                                          PAGE     28
                                      GLOSSARY BY LEVEL NUMBER FOR                                    GLOSSARY PAGE      1
                                      TRANSACTION GROUP - ORDENTRY

DATE CATALOGED - 02/11/02 14:27:06                          DATE LAST UPDATED - 02/11/02 14:27:06
CATALOGED BY -                                              LAST UPDATED BY -
EXPIRATION DATE -                                           DATE LAST USED -
MASTER FILE NAME = CUSTMAST                                 RECORD FORMAT = FIXED BLOCKED
NUMBER OF IDENTIFIERS IN GROUP = 5                          RECORD SIZE = 80
NUMBER OF ACTIONS IN GROUP = 23                             BLOCK SIZE = 800

                                    ************************************
                                    *  1  OF  5  IDENTIFIERS IN GROUP  *
                                    ************************************
**********    ACTIONS IN IDENTIFIER = 8          IDENTIFIER ONE = (3ADD)          IDENTIFIER TWO = (ORDR)
*3ADDORDR*    DEFAULT CREATE = NO                IDENTIFIER LOCATION = 1          IDENTIFIER LOCATION = 5
**********    DEFAULT INSERT = NO                IDENTIFIER LENGTH = 4            IDENTIFIER LENGTH = 4
                                                 IDENTIFIED IF EQUAL              IDENTIFIED IF EQUAL
-------------------------------------------------------------------------------------------------------------------------------
MASTER  .....TRANSACTION DATA.....
FILE    FIELD     FIELD     FIELD     DECIMAL   ACTION   LEVEL          MINIMUM                  MAXIMUM
FIELD   TYPE      LOCATION  LENGTH    PLACES    CODE     NUMBER         VALUE                    VALUE
-------------------------------------------------------------------------------------------------------------------------------
CUSTNO    C          9         5                M        1          EDIT = Y9999
ORDERNO   C         14         5                I        2
ORDRDATE  C         19         6                R        2                    DATE VALIDATION
ORPERSON  C         25         3                R        2          EDIT = -9-9-9
ORDPONUM  C         28         5                R        2
ORDDUDAT  C         33         6                R        2                    DATE VALIDATION
ORDINVGN  C         39         1                R        2          EDIT = Z
ORDCMPLT  C         40         1                R        2
FEB 11, 2002  14.27.06                                                                                          PAGE     29
                                      GLOSSARY BY LEVEL NUMBER FOR                                    GLOSSARY PAGE      2
                                      TRANSACTION GROUP - ORDENTRY

                                    ************************************
                                    *  2  OF  5  IDENTIFIERS IN GROUP  *
                                    ************************************
**********    ACTIONS IN IDENTIFIER = 6          IDENTIFIER ONE = (4ADD)          IDENTIFIER TWO = (ITEM)
*4ADDITEM*    DEFAULT CREATE = NO                IDENTIFIER LOCATION = 1          IDENTIFIER LOCATION = 5
**********    DEFAULT INSERT AT LEVEL 3          IDENTIFIER LENGTH = 4            IDENTIFIER LENGTH = 4
                                                 IDENTIFIED IF EQUAL              IDENTIFIED IF EQUAL
-------------------------------------------------------------------------------------------------------------------------------
MASTER  .....TRANSACTION DATA.....
FILE    FIELD     FIELD     FIELD     DECIMAL   ACTION   LEVEL          MINIMUM                  MAXIMUM
FIELD   TYPE      LOCATION  LENGTH    PLACES    CODE     NUMBER         VALUE                    VALUE
-------------------------------------------------------------------------------------------------------------------------------
CUSTNO    C          9         5                M        1          EDIT = Y9999
ORDERNO   C         14         5                M        2
ITEMORD   C         19         7                M        3
ITEMNAME  C         26        20                R        3
ITMPRICE  Z         47         6       2        R        3      ***0.00          ***      ***300.0         ***
ITMQTYOR  Z         60         7                R        3
FEB 11, 2002  14.27.06                                                                                          PAGE     30
                                      GLOSSARY BY LEVEL NUMBER FOR                                    GLOSSARY PAGE      3
                                      TRANSACTION GROUP - ORDENTRY

                                    ************************************
                                    *  3  OF  5  IDENTIFIERS IN GROUP  *
                                    ************************************
**********    ACTIONS IN IDENTIFIER = 4          IDENTIFIER ONE = (5ADD)          IDENTIFIER TWO = (ITEM)
*5ADDITEM*    DEFAULT CREATE = NO                IDENTIFIER LOCATION = 1          IDENTIFIER LOCATION = 5
**********    DEFAULT INSERT AT LEVEL 3          IDENTIFIER LENGTH = 4            IDENTIFIER LENGTH = 4
                                                 IDENTIFIED IF EQUAL              IDENTIFIED IF EQUAL
-------------------------------------------------------------------------------------------------------------------------------
MASTER  .....TRANSACTION DATA.....
FILE    FIELD     FIELD     FIELD     DECIMAL   ACTION   LEVEL          MINIMUM                  MAXIMUM
```

Figure D-67 Definition Run: ORDENTRY Transaction Group Definition (Page 2 of 3)

```
FIELD   TYPE    LOCATION   LENGTH   PLACES    CODE    NUMBER          VALUE                    VALUE
-------------------------------------------------------------------------------------------------------------
CUSTNO    C          9        5               M        1          EDIT = Y9999
ORDERNO   C         14        5               M        2
ITEMORD   C         19        7               M        3
ITMQTYOR  Z         26        5               R        3
FEB 11, 2002  14.27.06                                                                          PAGE     31
                                          GLOSSARY BY LEVEL NUMBER FOR                    GLOSSARY PAGE     4
                                          TRANSACTION GROUP - ORDENTRY

                                       ************************************
                                       *  4 OF 5 IDENTIFIERS IN GROUP   *
                                       ************************************
**********   ACTIONS IN IDENTIFIER = 2          IDENTIFIER ONE = (1DEL)          IDENTIFIER TWO = (ORDR)
*1DELORDR*   DEFAULT CREATE = NO                IDENTIFIER LOCATION = 1          IDENTIFIER LOCATION = 5
**********   DEFAULT INSERT = NO                IDENTIFIER LENGTH = 4            IDENTIFIER LENGTH = 4
                                                IDENTIFIED IF EQUAL              IDENTIFIED IF EQUAL
-------------------------------------------------------------------------------------------------------------
MASTER  .........TRANSACTION DATA.........
FILE    FIELD     FIELD     FIELD    DECIMAL   ACTION   LEVEL          MINIMUM                MAXIMUM
FIELD   TYPE      LOCATION  LENGTH   PLACES    CODE     NUMBER         VALUE                  VALUE
-------------------------------------------------------------------------------------------------------------
CUSTNO    C          9        5               M        1          EDIT = Y9999
ORDERNO   C         14        5               E        2
FEB 11, 2002  14.27.06                                                                          PAGE     32
                                          GLOSSARY BY LEVEL NUMBER FOR                    GLOSSARY PAGE     5
                                          TRANSACTION GROUP - ORDENTRY

                                       ************************************
                                       *  5 OF 5 IDENTIFIERS IN GROUP   *
                                       ************************************
**********   ACTIONS IN IDENTIFIER = 3          IDENTIFIER ONE = (2DEL)          IDENTIFIER TWO = (ITEM)
*2DELITEM*   DEFAULT CREATE = NO                IDENTIFIER LOCATION = 1          IDENTIFIER LOCATION = 5
**********   DEFAULT INSERT = NO                IDENTIFIER LENGTH = 4            IDENTIFIER LENGTH = 4
                                                IDENTIFIED IF EQUAL              IDENTIFIED IF EQUAL
-------------------------------------------------------------------------------------------------------------
MASTER  .........TRANSACTION DATA.........
FILE    FIELD     FIELD     FIELD    DECIMAL   ACTION   LEVEL          MINIMUM                MAXIMUM
FIELD   TYPE      LOCATION  LENGTH   PLACES    CODE     NUMBER         VALUE                  VALUE
-------------------------------------------------------------------------------------------------------------
CUSTNO    C          9        5               M        1          EDIT = Y9999
ORDERNO   C         14        5               M        2
ITEMORD   C         19        7               E        3
** CLSGT00  TYPE 0  END OF TRANSACTION DEFINITION GLOSSARY.                                                *****
```

Figure D-67 Definition Run: ORDENTRY Transaction Group Definition (Page 3 of 3)

## HDTRMSTR File Definition

```
FEB 11, 2002  14.27.06                                                                          PAGE     33
 ** CLSAA00  TYPE 0  DEFINITION HDTRMSTR IS BEING PROCESSED.                                            *****
-------------------------------------------------------------------------------------------------------------
   FILE    STMT    FILE    DELETE GLOSSARY RECORD RECORD RECORDS BUFFER MAPPED     MAPPING        MAPPED      SEQUENCE   LIST
   NAME    TYPE    IDENT    FILE   LEVEL   FORMAT  SIZE  PER BLK  SIZE   FILE    INIT     TERM    RECSIZ      73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------
(HDTRMSTR) (FD)              (R)             (V)                 (32760)                                            271
-------------------------------------------------------------------------------------------------------------
   FILE    STMT  SEGMENT  DEL SGMT VRT            SUP SEG  KEY              AUTH          EXTERNAL          SEQUENCE   LIST
   NAME    TYPE   NAME    ETE NMBR SEG            DPL KEY CHAR              ID          TABLE NAME          73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------
(HDTRMSTR) (LS) (HEADER )    ( 1)                                                                                   272
-------------------------------------------------------------------------------------------------------------
   FILE    STMT   FIELD    DEL SGMT LEV FIELD  FIELD FLD SEG FLD DEC CNT OCRS OUTP EDIT     DATA BASE NAME/       SEQUENCE   LIST
   NAME    TYPE   NAME    ETE NMBR NUM LOCATN LNGTH TYP KEY RND PLC SEG NTIM EDIT LNGH     COLUMN HEADING        73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------
(HDTRMSTR) (L0) (CUSTNO )    ( 1) (1) (   1) (  5) (C) (1)                                                          273
(HDTRMSTR) (L1) (CUSTNO )                                                          (CUSTOMER                )      274
(HDTRMSTR) (L2) (CUSTNO )                                                          (NUMBER                  )      275
(HDTRMSTR) (L0) (CUSTDATA)   ( 1) (1) (   6) ( 30) (C)                                                            276
(HDTRMSTR) (L1) (CUSTDATA)                                                         (CUSTOMER                )      277
(HDTRMSTR) (L2) (CUSTDATA)                                                         (DATA                    )      278
(HDTRMSTR) (L0) (COUNTA )    ( 1) (1) (  36) (  4) (F)           ( 2)                                              279
(HDTRMSTR) (L1) (COUNTA )                                                          (COUNT                   )      280
(HDTRMSTR) (L2) (COUNTA )                                                          (TRAILER A               )      281
(HDTRMSTR) (L0) (COUNTB )    ( 1) (1) (  40) (  4) (F)           ( 3)                                              282
(HDTRMSTR) (L1) (COUNTB )                                                          (COUNT                   )      283
(HDTRMSTR) (L2) (COUNTB )                                                          (TRAILER B               )      284
-------------------------------------------------------------------------------------------------------------
   FILE    STMT  SEGMENT  DEL SGMT VRT            SUP SEG  KEY              AUTH          EXTERNAL          SEQUENCE   LIST
   NAME    TYPE   NAME    ETE NMBR SEG            DPL KEY CHAR              ID          TABLE NAME          73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------
(HDTRMSTR) (LS) (TRAILERA)   ( 2)                                                                                   285
```

Figure D-68 Definition Run: HDTRMSTR File Definition (Page 1 of 3)

```
------------------------------------------------------------------------------------------------------------------------------------
   FILE    STMT   FIELD   DEL SGMT LEV FIELD  FIELD FLD SEG FLD DEC CNT OCRS OUTP EDIT     DATA BASE NAME/          SEQUENCE   LIST
   NAME    TYPE   NAME    ETE NMBR NUM LOCATN LNGTH TYP KEY RND PLC SEG NTIM EDIT LNGH     COLUMN HEADING           73 TO 80    SEQ
------------------------------------------------------------------------------------------------------------------------------------
(HDTRMSTR) (L0) (ORDERNO )    ( 2) (2) (   1) (  5) (C) (1)                                                                    286
(HDTRMSTR) (L1) (ORDERNO )                                                          (ORDER                    )              287
(HDTRMSTR) (L2) (ORDERNO )                                                          (NUMBER                   )              288
(HDTRMSTR) (L0) (ORDRDATA)    ( 2) (2) (   6) ( 17) (C)                                                                        289
(HDTRMSTR) (L1) (ORDRDATA)                                                          (ORDER                    )              290
(HDTRMSTR) (L2) (ORDRDATA)                                                          (DATA                     )              291
------------------------------------------------------------------------------------------------------------------------------------
   FILE    STMT  SEGMENT  DEL SGMT VRT           SUP SEG  KEY              AUTH         EXTERNAL              SEQUENCE   LIST
   NAME    TYPE   NAME    ETE NMBR SEG           DPL KEY CHAR              ID          TABLE NAME             73 TO 80    SEQ
------------------------------------------------------------------------------------------------------------------------------------
(HDTRMSTR) (LS) (TRAILERB)    ( 3)                                                                                             292
------------------------------------------------------------------------------------------------------------------------------------
   FILE    STMT   FIELD   DEL SGMT LEV FIELD  FIELD FLD SEG FLD DEC CNT OCRS OUTP EDIT     DATA BASE NAME/          SEQUENCE   LIST
   NAME    TYPE   NAME    ETE NMBR NUM LOCATN LNGTH TYP KEY RND PLC SEG NTIM EDIT LNGH     COLUMN HEADING           73 TO 80    SEQ
------------------------------------------------------------------------------------------------------------------------------------
(HDTRMSTR) (L0) (INSTNO  )    ( 3) (2) (   1) (  6) (C) (1)                                                                    293
(HDTRMSTR) (L1) (INSTNO  )                                                          (INSTALL                  )              294
(HDTRMSTR) (L2) (INSTNO  )                                                          (NUMBER                   )              295
(HDTRMSTR) (L0) (INSTDATA)    ( 3) (2) (   7) ( 23) (C)                                                                        296
(HDTRMSTR) (L1) (INSTDATA)                                                          (INSTALL                  )              297
(HDTRMSTR) (L2) (INSTDATA)                                                          (DATA                     )              298
** CLSDC02  TYPE 0  THE DEFINITION IS VALID AND HAS BEEN CATALOGED.                                                              *****
FEB 11, 2002  14.27.06                                                                                          PAGE      34
                                          FILE STRUCTURE DIAGRAM FOR                                      DIAGRAM PAGE  1
                                          FILE DEFINITION - HDTRMSTR

RECORD FORMAT = VARIABLE BLOCKED                                          FILE IDENTIFICATION =
RECORD SIZE = 32752   BLOCK SIZE = 32760                                  NUMBER OF SEGMENTS IN FILE = 3

                                              +---------+
                                              °HEADER   °
                                              +---------+
                                                   °
                                             +-----------+
                                             °           °
                                        +---------+ +---------+
                                        °TRAILERA ° °TRAILERB °
                                        +---------+ +---------+

FEB 11, 2002  14.27.06                                                                                          PAGE      35
                                     DETAILED GLOSSARY BY NAME FOR                                        GLOSSARY PAGE      1
                                       FILE DEFINITION - HDTRMSTR

DATE CATALOGED - 02/25/02 18:15:33                                        DATE LAST UPDATED - 02/25/02 18:15:33
CATALOGED BY -                                                            LAST UPDATED BY -
EXPIRATION DATE -                                                         DATE LAST USED -
FILE IDENTIFICATION =                                                     RECORD FORMAT = VARIABLE BLOCKED
NUMBER OF SEGMENTS IN FILE = 3                                            RECORD SIZE = 32752
NUMBER OF FIELDS IN FILE = 8                                              BLOCK SIZE = 32760

                                     ********************************
                                     *    SEGMENT   1, LEVEL 1      *
                                     *   SEGMENT NAME = HEADER      *
                                     ********************************
SEGMENT OCCURS N TIMES = 1                                  KEY FIELD 1 = CUSTNO     TYPE = C   LENGTH = 5
SEGMENT SIZE = 43
NUMBER OF FIELDS IN SEGMENT = 4
------------------------------------------------------------------------------------------------------------------------------------
FIELD   FIELD   FIELD    FIELD   FIELD    DEC    CNT FIELD  EDIT CODES     EDIT    OUTPUT   LINE  DATA BASE NAME/
NAME    TYPE   LOCATION  LENGTH  RNDING  PLACES  FOR SGMT   ( ) ( ) ( )   LENGTH   WIDTH    NO  *** COLUMN HEADING ***
------------------------------------------------------------------------------------------------------------------------------------
COUNTA    F        36        4                       2                      14       14
                                                                                          1   ***     COUNT      ***
                                                                                          2   ***   TRAILER A    ***
COUNTB    F        40        4                       3                      14       14
                                                                                          1   ***     COUNT      ***
                                                                                          2   ***   TRAILER B    ***
CUSTDATA  C         6       30                                              30       30
                                                                                          1   ***   CUSTOMER     ***
                                                                                          2   ***     DATA       ***
CUSTNO    C         1        5                                               5        8
                                                                                          1   ***   CUSTOMER     ***
                                                                                          2   ***    NUMBER      ***
FEB 11, 2002  14.27.06                                                                                          PAGE      36
                                     DETAILED GLOSSARY BY NAME FOR                                        GLOSSARY PAGE      2
                                       FILE DEFINITION - HDTRMSTR

                                     ********************************
                                     *    SEGMENT   2, LEVEL 2      *
                                     *   SEGMENT NAME = TRAILERA    *
                                     ********************************
COUNT FIELD FOR SEGMENT = COUNTA                            KEY FIELD 1 = ORDERNO    TYPE = C   LENGTH = 5
SEGMENT SIZE = 22
NUMBER OF FIELDS IN SEGMENT = 2
------------------------------------------------------------------------------------------------------------------------------------
```

Figure D-68 Definition Run: HDTRMSTR File Definition (Page 2 of 3)

```
FIELD   FIELD   FIELD    FIELD   FIELD    DEC    CNT FIELD  EDIT CODES     EDIT    OUTPUT   LINE  DATA BASE NAME/
NAME    TYPE   LOCATION  LENGTH  RNDING  PLACES  FOR SGMT   ( ) ( ) ( )   LENGTH  WIDTH     NO   *** COLUMN HEADING ***
-------------------------------------------------------------------------------------------------------------------------
ORDERNO   C        1       5                                                 5       6
                                                                                      1   ***     ORDER      ***
                                                                                      2   ***     NUMBER     ***
ORDRDATA  C        6      17                                                17      17
                                                                                      1   ***     ORDER      ***
                                                                                      2   ***     DATA       ***
FEB 11, 2002  14.27.06                                                                                    PAGE     37
                                    DETAILED GLOSSARY BY NAME FOR                                  GLOSSARY PAGE      3
                                      FILE DEFINITION - HDTRMSTR

                                   *********************************
                                   *    SEGMENT   3, LEVEL 2     *
                                   *   SEGMENT NAME = TRAILERB    *
                                   *********************************
COUNT FIELD FOR SEGMENT = COUNTB                                  KEY FIELD 1 = INSTNO     TYPE = C   LENGTH = 6
SEGMENT SIZE = 29
NUMBER OF FIELDS IN SEGMENT = 2
-------------------------------------------------------------------------------------------------------------------------
FIELD   FIELD   FIELD    FIELD   FIELD    DEC    CNT FIELD  EDIT CODES     EDIT    OUTPUT   LINE  DATA BASE NAME/
NAME    TYPE   LOCATION  LENGTH  RNDING  PLACES  FOR SGMT   ( ) ( ) ( )   LENGTH  WIDTH     NO   *** COLUMN HEADING ***
-------------------------------------------------------------------------------------------------------------------------
INSTDATA  C        7      23                                                23      23
                                                                                      1   ***    INSTALL     ***
                                                                                      2   ***     DATA       ***
INSTNO    C        1       6                                                 6       7
                                                                                      1   ***    INSTALL     ***
                                                                                      2   ***    NUMBER      ***
** CLSGB00  TYPE 0  END OF FILE DEFINITION GLOSSARY.                                                          *****
```

Figure D-68 Definition Run: HDTRMSTR File Definition (Page 3 of 3)

## HDTRATRN File Definition

```
FEB 11, 2002  14.27.06                                                                                    PAGE     38
 ** CLSAA00  TYPE 0  DEFINITION HDTRATRN IS BEING PROCESSED.                                                *****
-------------------------------------------------------------------------------------------------------------------------
   FILE    STMT    FILE    DELETE GLOSSARY RECORD RECORD RECORDS BUFFER MAPPED     MAPPING         MAPPED          SEQUENCE   LIST
   NAME    TYPE    IDENT    FILE   LEVEL   FORMAT  SIZE  PER BLK  SIZE  FILE    INIT      TERM    RECSIZ          73 TO 80   SEQ
-------------------------------------------------------------------------------------------------------------------------
(HDTRATRN) (FD)             (R)             (F)    (  80)                                                            299
-------------------------------------------------------------------------------------------------------------------------
   FILE    STMT  SEGMENT  DEL SGMT VRT            SUP SEG  KEY              AUTH          EXTERNAL            SEQUENCE   LIST
   NAME    TYPE   NAME    ETE NMBR SEG            DPL KEY CHAR              ID          TABLE NAME            73 TO 80   SEQ
-------------------------------------------------------------------------------------------------------------------------
(HDTRATRN) (LS) (ROOT    )    (01)                                                                                  300
-------------------------------------------------------------------------------------------------------------------------
   FILE    STMT   FIELD   DEL SGMT LEV FIELD  FIELD FLD SEG FLD DEC CNT OCRS OUTP EDIT      DATA BASE NAME/          SEQUENCE   LIST
   NAME    TYPE   NAME    ETE NMBR NUM LOCATN LNGTH TYP KEY RND PLC SEG NTIM EDIT LNGH      COLUMN HEADING           73 TO 80   SEQ
-------------------------------------------------------------------------------------------------------------------------
(HDTRATRN) (L0) (CUSTNO  )    (01) (1) (   1) (  5) (C)                                                             301
(HDTRATRN) (D1) (CUSTNO  )    (CUSTOMER NUMBER IN HEADER RECORD ONLY   )                                            302
(HDTRATRN) (L1) (CUSTNO  )                                                    (CUSTOMER                 )           303
(HDTRATRN) (L2) (CUSTNO  )                                                    (NUMBER                   )           304
(HDTRATRN) (L0) (CUSTDATA)    (01) (1) (   6) ( 30) (C)                                                             305
(HDTRATRN) (L1) (CUSTDATA)                                                    (CUSTOMER                 )           306
(HDTRATRN) (L2) (CUSTDATA)                                                    (DATA                     )           307
(HDTRATRN) (L0) (TRAILID )    (01) (1) (   1) (  1) (C)                                                             308
(HDTRATRN) (L1) (TRAILID )                                                    (TRAILER                  )           309
(HDTRATRN) (L2) (TRAILID )                                                    (RCD ID                   )           310
(HDTRATRN) (L0) (ORDRERNO)    (01) (1) (   2) (  5) (C)                                                             311
(HDTRATRN) (L1) (ORDRERNO)                                                    (ORDER                    )           312
(HDTRATRN) (L2) (ORDRERNO)                                                    (NUMBER                   )           313
(HDTRATRN) (L0) (ORDRDATA)    (01) (1) (   7) ( 17) (C)                                                             314
(HDTRATRN) (L1) (ORDRDATA)                                                    (ORDER                    )           315
(HDTRATRN) (L2) (ORDRDATA)                                                    (DATA                     )           316
(HDTRATRN) (L0) (INSTNO  )    (01) (1) (   2) (  6) (C)                                                             317
(HDTRATRN) (L1) (INSTNO  )                                                    (INSTALL                  )           318
(HDTRATRN) (L2) (INSTNO  )                                                    (NUMBER                   )           319
(HDTRATRN) (L0) (INSTDATA)    (01) (1) (   8) ( 23) (C)                                                             320
(HDTRATRN) (L1) (INSTDATA)                                                    (INSTALL                  )           321
(HDTRATRN) (L2) (INSTDATA)                                                    (DATA                     )           322
(HDTRATRN) (L0) (COMMCUST)    (01) (1) (  50) (  5) (C) (1)                                                         323
(HDTRATRN) (D1) (COMMCUST)    (COMMON CUST NO -  FROM HEADER RECORD    )                                            324
(HDTRATRN) (L1) (COMMCUST)                                                    (SAVED                    )           325
(HDTRATRN) (L2) (COMMCUST)                                                    (CUST NO                  )           326
(HDTRATRN) (L0) (ALL80   )    ( 1) (1) (   1) ( 80) (C)                                                             327
(HDTRATRN) (L1) (ALL80   )                                                    (TRAN RECORD              )           328
(HDTRATRN) (L2) (ALL80   )                                                    (ALL 80 POSITIONS         )           329
** CLSUN00  TYPE 1  SEGMENT   1 LEVEL 1 CONTAINS OVERLAPPING FIELDS.                                              *****
```

Figure D-69 Definition Run: HDTRATRN File Definition (Page 1 of 2)

```
                ALL80    OVERLAPS CUSTNO   TRAILID  INSTNO   ORDRERNO                                                *****
                                  CUSTDATA ORDRDATA INSTDATA COMMCUST                                                *****
                CUSTNO   OVERLAPS TRAILID  INSTNO   ORDRERNO                                                        *****
                INSTNO   OVERLAPS ORDRERNO CUSTDATA ORDRDATA                                                        *****
                ORDRERNO OVERLAPS CUSTDATA                                                                          *****
                CUSTDATA OVERLAPS ORDRDATA INSTDATA                                                                 *****
                ORDRDATA OVERLAPS INSTDATA                                                                          *****
** CLSDC02  TYPE 0  THE DEFINITION IS VALID AND HAS BEEN CATALOGED.                                                  *****
FEB 11, 2002  14.27.06                                                                              PAGE     39
                                        DETAILED GLOSSARY BY NAME FOR                          GLOSSARY PAGE      1
                                          FILE DEFINITION - HDTRATRN

DATE CATALOGED - 02/25/02 18:15:33                      DATE LAST UPDATED - 02/25/02 18:15:33
CATALOGED BY -                                          LAST UPDATED BY -
EXPIRATION DATE -                                       DATE LAST USED -
FILE IDENTIFICATION =                                   RECORD FORMAT = FIXED UNBLOCKED
NUMBER OF SEGMENTS IN FILE = 1                          RECORD SIZE = 80
NUMBER OF FIELDS IN FILE = 9                            BLOCK SIZE = 80

                                      ********************************
                                      *    SEGMENT   1, LEVEL 1      *
                                      *   SEGMENT NAME = ROOT        *
                                      ********************************

SEGMENT OCCURS N TIMES = 1                              KEY FIELD 1 = COMMCUST   TYPE = C   LENGTH = 5
SEGMENT SIZE = 80
NUMBER OF FIELDS IN SEGMENT = 9
-------------------------------------------------------------------------------------------------------------------
FIELD   FIELD   FIELD    FIELD   FIELD    DEC    CNT FIELD  EDIT CODES     EDIT    OUTPUT   LINE  DATA BASE NAME/
NAME    TYPE   LOCATION  LENGTH  RNDING  PLACES  FOR SGMT   ( ) ( ) ( )  LENGTH   WIDTH     NO  *** COLUMN HEADING ***
-------------------------------------------------------------------------------------------------------------------
ALL80     C        1       80                                              80      80
                                                                                      1   ***  TRAN RECORD   ***
                                                                                      2   ***ALL 80 POSITIONS***
COMMCUST  C       50        5                                               5       7
       DESCRIPTION: COMMON CUST NO -  FROM HEADER RECORD                              1   ***    SAVED      ***
                                                                                      2   ***    CUST NO    ***
CUSTDATA  C        6       30                                              30      30
                                                                                      1   ***    CUSTOMER   ***
                                                                                      2   ***      DATA     ***
CUSTNO    C        1        5                                               5       8
       DESCRIPTION: CUSTOMER NUMBER IN HEADER RECORD ONLY                             1   ***    CUSTOMER   ***
                                                                                      2   ***    NUMBER     ***
INSTDATA  C        8       23                                              23      23
                                                                                      1   ***    INSTALL    ***
                                                                                      2   ***    DATA       ***
INSTNO    C        2        6                                               6       7
                                                                                      1   ***    INSTALL    ***
                                                                                      2   ***    NUMBER     ***
ORDRDATA  C        7       17                                              17      17
                                                                                      1   ***    ORDER      ***
                                                                                      2   ***    DATA       ***
ORDRERNO  C        2        5                                               5       6
                                                                                      1   ***    ORDER      ***
                                                                                      2   ***    NUMBER     ***
FEB 11, 2002  14.27.06                                                                              PAGE     40
                                        DETAILED GLOSSARY BY NAME FOR                          GLOSSARY PAGE      2
                                          FILE DEFINITION - HDTRATRN
                                        SEGMENT   1, LEVEL 1 (CONTINUED)
-------------------------------------------------------------------------------------------------------------------
FIELD   FIELD   FIELD    FIELD   FIELD    DEC    CNT FIELD  EDIT CODES     EDIT    OUTPUT   LINE  DATA BASE NAME/
NAME    TYPE   LOCATION  LENGTH  RNDING  PLACES  FOR SGMT   ( ) ( ) ( )  LENGTH   WIDTH     NO  *** COLUMN HEADING ***
-------------------------------------------------------------------------------------------------------------------
TRAILID   C        1        1                                               1       7
                                                                                      1   ***    TRAILER    ***
                                                                                      2   ***    RCD ID     ***
** CLSGB00  TYPE 0  END OF FILE DEFINITION GLOSSARY.                                                               *****
```

Figure D-69 Definition Run: HDTRATRN File Definition (Page 2 of 2)

## HDTRGRP Transaction Group Definition

```
FEB 11, 2002  14.27.06                                                                              PAGE     41
 ** CLSAA00  TYPE 0  DEFINITION HDTRGRP  IS BEING PROCESSED.                                                       *****
-------------------------------------------------------------------------------------------------------------------
  GROUP    STMT IDENT1 IDENT1 ID1 IDENT2 IDENT2 ID2 DEL GLS    FILE    DEFLT DEFLT REC RECORD RECORDS BUFFER        SEQUENCE   LIST
  NAME     TYPE  CODE  LOCATN LNG  CODE  LOCATN LNG ETE ARY    NAME    CREAT INSRT FRM  SIZE  PER BLK  SIZE        73 TO 80   SEQ
-------------------------------------------------------------------------------------------------------------------
(HDTRGRP ) (TD) (A    ) (   1) (A) (B    ) (   1) (A) (R)      (HDTRMSTR)             (F) (  80)                          330
-------------------------------------------------------------------------------------------------------------------
  GROUP    STMT IDENT1 IDENT2 DEL FIELD  FIELD FLD   FIELD    ACT DEC            EDIT PATTERN          VAL       SEQUENCE   LIST
  NAME     TYPE  CODE   CODE  ETE LOCATN LNGTH TYP   NAME     ION PLCS  MINIMUM VALUE     MAXIMUM VALUE  TYP       73 TO 80   SEQ
-------------------------------------------------------------------------------------------------------------------
(HDTRGRP ) (TL) (A    ) (B    )      (  50) (  5) (C) (CUSTNO  ) (C)                                                      331
```

Figure D-70 Definition Run: HDTRGRP Transaction Group Definition (Page 1 of 3)

```
(HDTRGRP ) (TL) (A   ) (B   )     (   6) ( 30) (C) (CUSTDATA) (R)                                                                    332
-------------------------------------------------------------------------------------------------------------------------------------
  GROUP    STMT IDENT1 IDENT1 ID1 IDENT2 IDENT2 ID2 DEL GLS    FILE    DEFLT DEFLT REC RECORD RECORDS BUFFER          SEQUENCE   LIST
  NAME     TYPE  CODE  LOCATN LNG  CODE  LOCATN LNG ETE ARY    NAME    CREAT INSRT FRM  SIZE  PER BLK  SIZE          73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------------
(HDTRGRP ) (TD) (A   ) (   1) (1)                  (R)      (HDTRMSTR)            (F) (  80)                                    333
-------------------------------------------------------------------------------------------------------------------------------------
  GROUP    STMT IDENT1 IDENT2 DEL FIELD  FIELD FLD   FIELD    ACT DEC            EDIT PATTERN           VAL        SEQUENCE   LIST
  NAME     TYPE  CODE   CODE  ETE LOCATN LNGTH TYP   NAME     ION PLCS  MINIMUM VALUE     MAXIMUM VALUE  TYP        73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------------
(HDTRGRP ) (TL) (A   )            (  50) (  5) (C) (CUSTNO  ) (M)                                                                334
(HDTRGRP ) (TL) (A   )            (   2) (  5) (C) (ORDERNO ) (I)                                                                335
(HDTRGRP ) (TL) (A   )            (   7) ( 17) (C) (ORDRDATA) (R)                                                                336
-------------------------------------------------------------------------------------------------------------------------------------
  GROUP    STMT IDENT1 IDENT1 ID1 IDENT2 IDENT2 ID2 DEL GLS    FILE    DEFLT DEFLT REC RECORD RECORDS BUFFER          SEQUENCE   LIST
  NAME     TYPE  CODE  LOCATN LNG  CODE  LOCATN LNG ETE ARY    NAME    CREAT INSRT FRM  SIZE  PER BLK  SIZE          73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------------
(HDTRGRP ) (TD) (B   ) (   1) (1)                  (R)      (HDTRMSTR)            (F) (  80)                                    337
-------------------------------------------------------------------------------------------------------------------------------------
  GROUP    STMT IDENT1 IDENT2 DEL FIELD  FIELD FLD   FIELD    ACT DEC            EDIT PATTERN           VAL        SEQUENCE   LIST
  NAME     TYPE  CODE   CODE  ETE LOCATN LNGTH TYP   NAME     ION PLCS  MINIMUM VALUE     MAXIMUM VALUE  TYP        73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------------
(HDTRGRP ) (TL) (B   )            (  50) (  5) (C) (CUSTNO  ) (M)                                                                338
(HDTRGRP ) (TL) (B   )            (   2) (  6) (C) (INSTNO  ) (I)                                                                339
(HDTRGRP ) (TL) (B   )            (   8) ( 23) (C) (INSTDATA) (R)                                                                340
** CLSTN00  TYPE 1  WARNING - IDENTIFIERS DO NOT ALL HAVE SAME LOCATION AND SAME LENGTH.                                          *****
** CLSDC02  TYPE 0  THE DEFINITION IS VALID AND HAS BEEN CATALOGED.                                                               *****
FEB 11, 2002  14.27.06                                                                                              PAGE     42
                                        GLOSSARY BY LEVEL NUMBER FOR                                        GLOSSARY PAGE      1
                                        TRANSACTION GROUP - HDTRGRP

DATE CATALOGED - 02/25/02 18:15:33                          DATE LAST UPDATED - 02/25/02 18:15:33
CATALOGED BY -                                              LAST UPDATED BY -
EXPIRATION DATE -                                           DATE LAST USED -
MASTER FILE NAME = HDTRMSTR                                 RECORD FORMAT = FIXED UNBLOCKED
NUMBER OF IDENTIFIERS IN GROUP = 3                          RECORD SIZE = 80
NUMBER OF ACTIONS IN GROUP = 8                              BLOCK SIZE = 80

                                      ************************************
                                      *  1 OF 3  IDENTIFIERS IN GROUP   *
                                      ************************************
**********    ACTIONS IN IDENTIFIER = 2          IDENTIFIER ONE = (A)            IDENTIFIER TWO = (B)
*A   B   *    DEFAULT CREATE = NO                IDENTIFIER LOCATION = 1         IDENTIFIER LOCATION = 1
**********    DEFAULT INSERT = NO                IDENTIFIER LENGTH = 1           IDENTIFIER LENGTH = 1
                                                 IDENTIFIED IF NOT EQUAL         IDENTIFIED IF NOT EQUAL
-------------------------------------------------------------------------------------------------------------------------------------
MASTER  ..........TRANSACTION DATA..........
FILE    FIELD     FIELD     FIELD    DECIMAL   ACTION   LEVEL          MINIMUM                MAXIMUM
FIELD   TYPE     LOCATION   LENGTH   PLACES     CODE    NUMBER          VALUE                  VALUE
-------------------------------------------------------------------------------------------------------------------------------------
CUSTNO    C          50        5                C        1
CUSTDATA  C           6       30                R        1
FEB 11, 2002  14.27.06                                                                                              PAGE     43
                                        GLOSSARY BY LEVEL NUMBER FOR                                        GLOSSARY PAGE      2
                                        TRANSACTION GROUP - HDTRGRP

                                      ************************************
                                      *  2 OF 3  IDENTIFIERS IN GROUP   *
                                      ************************************
**********    ACTIONS IN IDENTIFIER = 3          IDENTIFIER ONE = (A)
*A       *    DEFAULT CREATE = NO                IDENTIFIER LOCATION = 1
**********    DEFAULT INSERT = NO                IDENTIFIER LENGTH = 1
                                                 IDENTIFIED IF EQUAL
-------------------------------------------------------------------------------------------------------------------------------------
MASTER  ..........TRANSACTION DATA..........
FILE    FIELD     FIELD     FIELD    DECIMAL   ACTION   LEVEL          MINIMUM                MAXIMUM
FIELD   TYPE     LOCATION   LENGTH   PLACES     CODE    NUMBER          VALUE                  VALUE
-------------------------------------------------------------------------------------------------------------------------------------
CUSTNO    C          50        5                M        1
ORDERNO   C           2        5                I        2
ORDRDATA  C           7       17                R        2
FEB 11, 2002  14.27.06                                                                                              PAGE     44
                                        GLOSSARY BY LEVEL NUMBER FOR                                        GLOSSARY PAGE      3
                                        TRANSACTION GROUP - HDTRGRP

                                      ************************************
                                      *  3 OF 3  IDENTIFIERS IN GROUP   *
                                      ************************************
**********    ACTIONS IN IDENTIFIER = 3          IDENTIFIER ONE = (B)
*B       *    DEFAULT CREATE = NO                IDENTIFIER LOCATION = 1
**********    DEFAULT INSERT = NO                IDENTIFIER LENGTH = 1
                                                 IDENTIFIED IF EQUAL
-------------------------------------------------------------------------------------------------------------------------------------
MASTER  ..........TRANSACTION DATA..........
FILE    FIELD     FIELD     FIELD    DECIMAL   ACTION   LEVEL          MINIMUM                MAXIMUM
FIELD   TYPE     LOCATION   LENGTH   PLACES     CODE    NUMBER          VALUE                  VALUE
-------------------------------------------------------------------------------------------------------------------------------------
```

Figure D-70 Definition Run: HDTRGRP Transaction Group Definition (Page 2 of 3)

```
CUSTNO    C         50       5              M        1
INSTNO    C          2       6              I        2
INSTDATA  C          8      23              R        2
** CLSGT00  TYPE 0  END OF TRANSACTION DEFINITION GLOSSARY.                                                    *****
FEB 11, 2002  14.27.06                                                                                 PAGE     45
** MK4W204  TYPE 0  NUMBER OF MESSAGES PRINTED IS   52.                                                       *****
                   M4INPUT  -      340 RECORDS INPUT
                   M4LIST   -     1454 RECORDS OUTPUT
                     15 TRACKS ASSIGNED TO M4LIB     --   13 TRACKS NOT FULL. LIBRARY DIRECTORY BLOCKING FACTOR IS  800.
** MK4W209  TYPE 0   924904 BYTES OF MAIN STORAGE UNUSED DURING DICTIONARY MAINTENANCE                          *****
```

Figure D-70 Definition Run: HDTRGRP Transaction Group Definition (Page 3 of 3)

# ITEM Master Create Run

## Banner Page

```
FEB 11, 2002  14.27.10                                                                                 PAGE      1
                                          ******************************************
                                          * VISION:Builder   4400  (OS/390 - 14.0)  *
                                          *           COPYRIGHT (C) 2002           *
                                          *  COMPUTER ASSOCIATES INTERNATIONAL, INC. *
                                          ******************************************
VISION:Builder JCR1 14.0 020802              ENABLED FOR IBM LANGUAGE ENVIRONMENT              BUILD STAMP = 102039,18:29:48.
                                          COMLIB  MODIFICATIONS (PTFs, APARs) INSTALLED
                                       201,202,203,204,205,206,207,208,209,210,212,213,214,215,216
```

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| INSTALLATION PARAMETERS (M4PARAMS,MARKLIBP) | | | | | | | |
| SYSTEM DELIMITER: | # | PAGE HEIGHT: | 66 | M4LIST WIDTH: | 132 | DEF WIDTH OF PAGE: | 0 |
| AUTO GRAND: | N | HEADING CHAR: | - | SUBTITLE REPEAT: | N | INVALID FIELD: | * |
| MISSING FIELD: | - | NON-EDIT FIELD: | + | PERCENT CHAR: | % | LEFT SEPARATOR: | ( |
| RIGHT SEPARATOR: | ) | SINGLE SEPARATOR: | , | SOURCE SPACING: | 1 | PRINT MESSAGES: | Y |
| CONSOLE MESSAGES: | N | M4REPO BLOCKSIZE: | 4,096 | INPUT I/O BUFFERS: | 2 | OUTPUT I/O BUFFERS: | 1 |
| SNGL-STEP STORAGE: | 8,192 | SNGL-STEP SORTSIZE: | 524,288 | DIGIT SELECT CHAR: | 9 | ZERO SUPPRESS CHAR: | Z |
| CURRENCY CHAR: | $ | PLUS CHAR: | + | MINUS CHAR: | - | CHECK PROTECT CHAR: | * |
| DECIMAL CHAR: | . | GROUPING CHAR: | , | PRIMARY PLOT CHAR: | X | SECONDARY PLOT CHAR: | * |
| FIT PLOT CHAR: | . | HORIZONTAL AXIS: | _ | HORIZONTAL HASH: | \| | VERTICAL AXIS: | \| |
| VERTICAL HASH: | - | MINUTES/HOUR: | 60 | SECONDS/MINUTE: | 60 | TIME DELIMITER: | HH:MM:SS |
| DATE FORMAT: | MMM DD, YYYY | TODAY FORMAT+DELIM: | MM/DD/YY | ISDATE DELIMITER: | YYYY-MM-DD | JULIAN DELIMITER: | YY.DDD |
| SORT PROGRAM CODE: | 2 | MINCORE VALUE: | 12 K | ALT M4LIST WIDTH: | 264 | ALT DEF W/OF PAGE: | 0 |
| MAX LINES OF TRACE: | 1,024 | ITEM TRACKING: | 0 | SUPPRESS NDS REPT?: | N | DEFAULT MAXGETMN: | 1,024 K |
| CONDITION CODE 1: | 0 | CONDITION CODE 2: | 4 | CONDITION CODE 3: | 8 | CONDITION CODE 4: | 16 |
| ISAM INDEX INCORE: | N | M4LIB BLKG FACTOR: | 0 | M4LIB RESERVE: | 0 | M4LIB COMPONENT: | C4.5 B/V |
| DEFAULT F/P AMODE: | 24 | M4LIB COMPRESSION?: | N | M4LIB COMPRESS MIN: | 507 | | |

Figure D-71 ITEM Master Create Run: Banner Page

## Application Coding

```
FEB 11, 2002  14.27.10                                                                                                PAGE      2
-------------------------------------------------------------------------------------------------------------------------------
   RUN     STMT    FILE    O N S D U T R A B S D S L  RB  R S M R S            START           END             SEQUENCE   LIST
   NAME    TYPE    NAME    L E E I P R E U U R L C S  PL  J O O F S            SEARCH          SEARCH          73 TO 80   SEQ
                           D W Q R D N P D F T M N T  TK  T P P O R
-------------------------------------------------------------------------------------------------------------------------------
(BLDIFILE) (RC)  (ITEMDAT1) (S)          (S)        (T,G)          (R)                                      (ASLGEN'D)     1
    CONTROL NAME BLDIFILE TERM LISTGEN                                                                                       2
    FILE MASTER INPUT  ITEMDAT1                                                                                              3
    FILE REPORT                                                                                                              4
-------------------------------------------------------------------------------------------------------------------------------
   RUN     STMT  LOGICAL  LBL  DTF/DD     KEY       KEY       KEY      QUALIFIER   USER I/O  ARRAY       PASSWORD/  SEQUENCE   LIST
   NAME    TYPE  FILENAME       NAME     FIELD-1   FIELD-2   FIELD-3   ICF CRD DBI   MODULE   QUAL        AUTH.-ID   73 TO 80   SEQ
-------------------------------------------------------------------------------------------------------------------------------
(BLDIFILE) (RF)  (ITEMDAT1) (S) (M4OLD    )                                                                  (ASLGEN'D)     5


=============================================================================================================================
=                                                                                                                           =
=                                                F I L E    S U M M A R Y                                                   =
=                                                                                                                           =
= JOBNAME: KREJO04X                          STEPNAME: PROC1                                                                =
=                                                                                                                           =
=  FILE NAME: M4OLD                          DSNAME: KREJO04.KREJO04X.JOB01802.D0000103.?                                   =
=  FILE NAME: M4LIB                          DSNAME: BUILDER.R140.M4LIB.R15#1                                               =
=                                                                                                                           =
=============================================================================================================================
FEB 11, 2002  14.27.10                                                                                                PAGE      3
                                                  ******************************
                                                  *  PROC NAME - REPT__1     *
                                                  *  INPUT STREAM PROCEDURE  *
                                                  ******************************
-------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                              MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID         ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM      73 TO 80   SEQ
-------------------------------------------------------------------------------------------------------------------------------
(ER)                                                                    (N)                                       (ASLGEN'D)     6
    REPORT ITEMNO ITEMNAME ITMPRICE                                                                                          7
      TITLE 'BUILD THE ITEM MAST - A VSAM DATA SET'                                                                          8
      TITLE 'LIST THE DATA OUTPUT TO THE VSAM ITEM FILE'                                                                     9
    END REPORT                                                                                                              10
-------------------------------------------------------------------------------------------------------------------------------
 ST SUM V  8  PG  PG   I SP  MAXIMUM TB COLUMN  DT  PG  STRT LIN SL RH .......SUBFILE........ C E A S R S N      C  SEQUENCE   LIST
 TY RPT S LPI WID HGT  M FRM LPP PGS TO HEADING POS POS PAGE NUM UB PN ENT    NAME   BLK/ FMT H F G T S U D      O  73 TO 80   SEQ
 MP     P            G           LT TYP POS        NUM     ML TD REC          BUF    C C S L T M S      N
-------------------------------------------------------------------------------------------------------------------------------
(E1)                                                                                                          (ASLGEN'D)    11
-------------------------------------------------------------------------------------------------------------------------------
         SP  Q             E N  S  D  C  S  T  C  C  M  M  A  M  D Q PCT/RAT     P              PARTIAL   SEQUENCE   LIST
STMT SEQ BF  L  FIELD  SEG-  N P  E  E  T  U  O  U  N  A  I  V  O  E L  FIELD   RA- C     PICTURE  STRT NO  73 TO 80   SEQ
TYPE NO. COL F  NAME     LVL D R  Q  S  L  B  T  M  T  X  N  G  D  C F  NAME    TIO T      EDIT    CHR CHR
-------------------------------------------------------------------------------------------------------------------------------
(R1)          (ITEMNO  ) 001-1                                                                                (ASLGEN'D)    12
(R1)          (ITEMNAME) 001-1                                                                                (ASLGEN'D)    13
(R1)          (ITMPRICE) 001-1                                                                                (ASLGEN'D)    14
-------------------------------------------------------------------------------------------------------------------------------
STMT  SEQ                                                                                                      SEQUENCE   LIST
TYPE  NO.                                        TEXT                                                          73 TO 80   SEQ
-------------------------------------------------------------------------------------------------------------------------------
(T1) (101)              (BUILD THE ITEM MAST - A VSAM DATA SET#                 )                              (ASLGEN'D)    15
(T1) (102)              (LIST THE DATA OUTPUT TO THE VSAM ITEM FILE#            )                              (ASLGEN'D)    16
FEB 11, 2002  14.27.10                                                                                                PAGE      4
                                                  ******************************
                                                  *  PROC NAME - SUBF__2     *
                                                  *  INPUT STREAM PROCEDURE  *
                                                  ******************************
-------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                              MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID         ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM      73 TO 80   SEQ
-------------------------------------------------------------------------------------------------------------------------------
(ER)                                                                    (N)                                       (ASLGEN'D)    17
    EXTRACT DDNAME ITMFILE RECFM KEYVSAM,                                                                                   18
            ITEMS ITEMNO ITEMNAME ITMPRICE                                                                                  19
-------------------------------------------------------------------------------------------------------------------------------
 ST SUM V  8  PG  PG   I SP  MAXIMUM TB COLUMN  DT  PG  STRT LIN SL RH .......SUBFILE........ C E A S R S N      C  SEQUENCE   LIST
 TY RPT S LPI WID HGT  M FRM LPP PGS TO HEADING POS POS PAGE NUM UB PN ENT    NAME   BLK/ FMT H F G T S U D      O  73 TO 80   SEQ
 MP     P            G           LT TYP POS        NUM     ML TD REC          BUF    C C S L T M S      N
-------------------------------------------------------------------------------------------------------------------------------
(E1)                                                          (DO)   (ITMFILE )     (K)                       (ASLGEN'D)    20
-------------------------------------------------------------------------------------------------------------------------------
         SP  Q             E N  S  D  C  S  T  C  C  M  M  A  M  D Q PCT/RAT     P              PARTIAL   SEQUENCE   LIST
STMT SEQ BF  L  FIELD  SEG-  N P  E  E  T  U  O  U  N  A  I  V  O  E L  FIELD   RA- C     PICTURE  STRT NO  73 TO 80   SEQ
```

Figure D-72 ITEM Master Create Run: Application Coding (Page 1 of 2)

```
TYPE NO.  COL F  NAME      LVL  D  R  Q  S  L  B  T  M  T  X  N  G  D  C F  NAME    TIO T       EDIT       CHR CHR
-------------------------------------------------------------------------------------------------------------------------------
(R1)           (ITEMNO  ) 001-1                                                                                (ASLGEN'D)    21
(R1)           (ITEMNAME) 001-1                                                                                (ASLGEN'D)    22
(R1)           (ITMPRICE) 001-1                                                                                (ASLGEN'D)    23
** MK4TY05  TYPE 0  SUBFILE RECORD LENGTH COMPUTED FROM THE ABOVE SPECIFICATION IS    33.                                 *****
FEB 11, 2002  14.27.10                                                                                          PAGE      5
** MK4W204  TYPE 0  NUMBER OF MESSAGES PRINTED IS    1.                                                                   *****
                    M4OLD    -       38 RECORDS INPUT
                    M4REPO   -       51 RECORDS OUTPUT
                    M4INPUT  -        9 RECORDS INPUT
                    M4LIST   -      140 RECORDS OUTPUT
                    ITMFILE  -       38 RECORDS OUTPUT
                      15 TRACKS ASSIGNED TO M4LIB    --   13 TRACKS NOT FULL. LIBRARY DIRECTORY BLOCKING FACTOR IS  800.
** MK4W209  TYPE 0  1017120 BYTES OF MAIN STORAGE UNUSED DURING DECODING PHASE                                          *****
                    1017608 BYTES OF MAIN STORAGE UNUSED DURING COMPILATION PHASE
                     564896 BYTES OF MAIN STORAGE UNUSED DURING PROCESSING PHASE
```

Figure D-72 ITEM Master Create Run: Application Coding (Page 2 of 2)

## Listing of the File Contents

```
** MK4ED02  TYPE 0  REQUEST-REPT___1  REPORT-1  REQUESTOR ID-                                                          *****
                                                  BUILD THE ITEM MAST - A VSAM DATA SET                              PAGE 1
                                                LIST THE DATA OUTPUT TO THE VSAM ITEM FILE
-------------------------------------------------------------------------------------------------------------------------------
   ITEM           ITEM           ITEM
  NUMBER          NAME           PRICE
-------------------------------------------------------------------------------------------------------------------------------
  D77      V:TRAN CLASS GDE-A      $97.50
  E56      V:BLDR USER I/O GDE      27.50
  E66      V:BLDR GDBI GUIDE        37.50
  F74      V:TRAN CLASS GDE-C      103.50
  F87      V:TRAN CLASS GDE-B       97.50
  F91      V:TRAN ADM GUIDE         30.00
  G38      1" BINDER NOTEBOOK        5.00
  G40      V:BLDR REFERENCE GDE     35.00
  G46      V:TRAN UTILITY GDE       15.00
  H61      V:BLDR INST GDE-VSE      15.00
  H98      OB6789A120481NN8         15.00
  K69      EEO REPORT GUIDE         67.00
  L06      V:INFM CLASS PACK       299.99
  L06F     V:INFM CLASS SLIDES      99.00
  L96      V:BLDR APPL GUIDE        50.00
  L97      V:INFM QUERY GUIDE       45.00
  L99      V:BLDR ENVIRON GDE-O     17.50
  M01      V:TRAN ENV GDE CICS      47.50
  M14      V:BLDR ASL REF GDE       31.00
  M23      V:TRAN SCREEN GDE        25.00
  M33      V:TRAN 7.5 CHG PAGES      5.00
  M50      V:TRAN SECURITY GDE      27.50
  M56      2" BINDER NOTEBOOK        7.50
  M57      3" BINDER NOTEBOOK        9.50
  010      ACCOUNTS PAYABLE UM      45.00
  013      V:BLDR REF SUMMARY       51.00
  019      V:BLDR ENV GUIDE-VSE     51.00
  027      V:BLDR CLASS GUIDE-A     55.00
  041      V:BLDR PROCEDURE GDE     55.00
  072      V:BLDR CLASS GUIDE-B     55.00
  106      V:BLDR TABLE TIPS         5.00
  110      V:BLDR REPORT TIPS        5.00
  852      PURCHASE ORDER ENVR      24.00
  865      V:INFM REFERENCE GDE     39.00
  866      V:INFM INSTALL GDE       19.00
  872      G/L REFERENCE GUIDE      75.00
  873      V:INFM OPS GUIDE         19.00
  978      PURCHASE ORDER GDE       35.00
** MK4FT03  TYPE 0  END OF REPORT.                                                                                      *****
** MK4W204  TYPE 0  NUMBER OF MESSAGES PRINTED IS    2.                                                                  *****
                    M4REPO   -       50 RECORDS INPUT
                    M4LIST   -       48 RECORDS OUTPUT
** MK4W209  TYPE 0  1005564 BYTES OF MAIN STORAGE UNUSED DURING REPORTING PHASE                                         *****
```

Figure D-73 ITEM Master Create Run: Listing of the File Contents

# CUSTMAST Create Run

## Banner Page

```
FEB 11, 2002  14.27.14                                                                                    PAGE      1

                                                ****************************************
                                                *  VISION:Builder   4400  (OS/390 - 14.0)  *
                                                *           COPYRIGHT (C) 2002           *
                                                *  COMPUTER ASSOCIATES INTERNATIONAL, INC. *
                                                ****************************************
 VISION:Builder JCR1 14.0 020802                    ENABLED FOR IBM LANGUAGE ENVIRONMENT                    BUILD STAMP = 102039,18:29:48.
                                                 COMLIB  MODIFICATIONS (PTFs, APARs) INSTALLED
                                          201,202,203,204,205,206,207,208,209,210,212,213,214,215,216

=============================================================================================================================
=                                                                                                                           =
=                    I N S T A L L A T I O N    P A R A M E T E R S    ( M 4 P A R A M S , M A R K L I B P )               =
=                                                                                                                           =
=  SYSTEM DELIMITER:   #          PAGE HEIGHT:       66        M4LIST WIDTH:      132       DEF WIDTH OF PAGE:  0         =
=                                                                                                                           =
=  AUTO GRAND:         N          HEADING CHAR:      -         SUBTITLE REPEAT:   N         INVALID FIELD:      *         =
=                                                                                                                           =
=  MISSING FIELD:      -          NON-EDIT FIELD:    +         PERCENT CHAR:      %         LEFT SEPARATOR:     (         =
=                                                                                                                           =
=  RIGHT SEPARATOR:    )          SINGLE SEPARATOR:  ,         SOURCE SPACING:    1         PRINT MESSAGES:     Y         =
=                                                                                                                           =
=  CONSOLE MESSAGES:   N          M4REPO BLOCKSIZE:  4,096     INPUT I/O BUFFERS: 2         OUTPUT I/O BUFFERS: 1         =
=                                                                                                                           =
=  SNGL-STEP STORAGE:  8,192      SNGL-STEP SORTSIZE: 524,288  DIGIT SELECT CHAR: 9         ZERO SUPPRESS CHAR: Z         =
=                                                                                                                           =
=  CURRENCY CHAR:      $          PLUS CHAR:         +         MINUS CHAR:        -         CHECK PROTECT CHAR: *         =
=                                                                                                                           =
=  DECIMAL CHAR:       .          GROUPING CHAR:     ,         PRIMARY PLOT CHAR: X         SECONDARY PLOT CHAR:*         =
=                                                                                                                           =
=  FIT PLOT CHAR:      .          HORIZONTAL AXIS:   _         HORIZONTAL HASH:   |         VERTICAL AXIS:     |         =
=                                                                                                                           =
=  VERTICAL HASH:      -          MINUTES/HOUR:      60        SECONDS/MINUTE:    60        TIME DELIMITER:     HH:MM:SS  =
=                                                                                                                           =
=  DATE FORMAT:      MM DD, YYYY  TODAY FORMAT+DELIM: MM/DD/YY  ISDATE DELIMITER:  YYYY-MM-DD JULIAN DELIMITER:  YY.DDD    =
=                                                                                                                           =
=  SORT PROGRAM CODE:  2          MINCORE VALUE:     12 K      ALT M4LIST WIDTH:  264       ALT DEF W/OF PAGE:  0         =
=                                                                                                                           =
=  MAX LINES OF TRACE: 1,024      ITEM TRACKING:     0         SUPPRESS NDS REPT?: N        DEFAULT MAXGETMN:   1,024 K   =
=                                                                                                                           =
=  CONDITION CODE 1:   0          CONDITION CODE 2:  4         CONDITION CODE 3:  8         CONDITION CODE 4:   16        =
=                                                                                                                           =
=  ISAM INDEX INCORE:  N          M4LIB BLKG FACTOR: 0         M4LIB RESERVE:     0         M4LIB COMPONENT:    C4.5 B/V  =
=                                                                                                                           =
=  DEFAULT F/P AMODE:  24         M4LIB COMPRESSION?: N        M4LIB COMPRESS MIN: 507                                    =
=                                                                                                                           =
=============================================================================================================================
```

Figure D-74 CUSTMAST Create Run: Banner Page

## Application Coding

```
FEB 11, 2002  14.27.14                                                                                    PAGE      2
-----------------------------------------------------------------------------------------------------------------------------
   RUN     STMT    FILE    O N S D U T R A B S D S L  RB  R S M R S          START          END           SEQUENCE   LIST
   NAME    TYPE    NAME    L E E I P R E U U R L C S  PL  J O O F S          SEARCH         SEARCH        73 TO 80    SEQ
                           D W Q R D N P D F T M N T  TK  T P P O R
-----------------------------------------------------------------------------------------------------------------------------
(BLDCUSTM) (RC) (CUSTMAST)  (S)     (S,S)       (T,G)         (R)                                        (ASLGEN'D)     1
    CONTROL NAME BLDCUSTM TERM LISTGEN                                                                                    2
    FILE MASTER OUTPUT CUSTMAST                                                                                           3
    FILE TRAN   TRANWORK GROUPS CUSTADDS ORDENTRY                                                                         4
    FILE REPORT                                                                                                           5
-----------------------------------------------------------------------------------------------------------------------------
   RUN     STMT  LOGICAL  LBL  DTF/DD     KEY       KEY       KEY     QUALIFIER   USER I/O  ARRAY      PASSWORD/  SEQUENCE   LIST
   NAME    TYPE  FILENAME       NAME     FIELD-1   FIELD-2   FIELD-3  ICF CRD DBI   MODULE   QUAL       AUTH.-ID   73 TO 80    SEQ
-----------------------------------------------------------------------------------------------------------------------------
(BLDCUSTM) (RF) (CUSTMAST) (S) (M4NEW    )                                                                 (ASLGEN'D)     6
(BLDCUSTM) (RF) (TRANWORK) (S) (M4TRAN   )                                                                 (ASLGEN'D)     7
-----------------------------------------------------------------------------------------------------------------------------
   RUN     STMT                TRANSACTION GROUPS                                                          SEQUENCE   LIST
   NAME    TYPE     1         2         3         4         5                                              73 TO 80    SEQ
-----------------------------------------------------------------------------------------------------------------------------
(BLDCUSTM) (RG) (CUSTADDS) (ORDENTRY)                                                                      (ASLGEN'D)     8
```

Figure D-75 CUSTMAST Create Run: Application Coding (Page 1 of 2)

```
=                                                                                                                                =
=                                                    F I L E   S U M M A R Y                                                     =
=                                                                                                                                =
= JOBNAME: KREJO04X                                 STEPNAME: PROC2                                                              =
=                                                                                                                                =
=  FILE NAME: M4NEW                                 DSNAME: BUILDER.R140.CUSTMAST.BUILT.R15#1                                    =
=  FILE NAME: M4TRAN                                DSNAME: KREJO04.KREJO04X.JOB01802.D0000105.?                                 =
=  FILE NAME: M4LIB                                 DSNAME: BUILDER.R140.M4LIB.R15#1                                             =
=                                                                                                                                =
================================================================================================================================

FEB 11, 2002  14.27.14                                                                                                  PAGE      3

                                                    ****************************
                                                    *  PROC NAME - REPT  1     *
                                                    *  INPUT STREAM PROCEDURE  *
                                                    ****************************

--------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                              MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID         ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM      73 TO 80    SEQ
--------------------------------------------------------------------------------------------------------------------------------
(ER)                                                                          (N)                                (ASLGEN'D)      9
    REPORT CUSTNO ORDERNO ORDRDATE ORPERSON ORDPONUM ORDDUDAT,                                                                    10
           ORDINVGN ORDCMPLT ITEMORD ITEMNAME ITMPRICE                                                                           11
      GROUP BY CUSTNO AT LEVEL 1                                                                                                 12
      GROUP BY ORDERNO ORDRDATE ORPERSON ORDPONUM,                                                                               13
               ORDDUDAT ORDINVGN ORDCMPLT AT LEVEL 2                                                                             14
      ITEM CUSTNO,                                                                                                               15
           SPACES 0                                                                                                              16
      ITEM ORDPONUM ORDDUDAT ORDINVGN ORDCMPLT,                                                                                  17
           ITEMORD ITEMNAME ITMPRICE,                                                                                            18
           SPACES 1                                                                                                              19
      TITLE 'BUILD THE CUSTOMER MASTER FILE FOR THE ADVANCED',                                                                   20
            ' TRANSACTIONS EXAMPLES'                                                                                             21
      TITLE 'LIST THE INITIALLY CREATED CUSTOMER MASTER'                                                                         22
    END REPORT                                                                                                                   23
--------------------------------------------------------------------------------------------------------------------------------
 ST SUM V  8  PG  PG   I SP  MAXIMUM TB COLUMN  DT  PG  STRT LIN SL RH .......SUBFILE..... C E A S R S N      C  SEQUENCE   LIST
 TY RPT S LPI WID HGT  M FRM LPP PGS TO HEADING POS POS PAGE NUM UB PN ENT    NAME   BLK/ FMT H F G T S U D      O  73 TO 80    SEQ
 MP     P            G            LT TYP POS         NUM      ML TD REC          BUF      C C S L T M S      N
--------------------------------------------------------------------------------------------------------------------------------
(E1)                                                                                                                 (ASLGEN'D)     24
--------------------------------------------------------------------------------------------------------------------------------
        SP  Q              E N S D C S T C C M M A M D Q PCT/RAT     P              PARTIAL   SEQUENCE   LIST
STMT SEQ BF  L  FIELD  SEG- N P E E T U O U N A I V O E L FIELD   RA- C     PICTURE     STRT NO   73 TO 80    SEQ
TYPE NO. COL F  NAME     LVL D R Q S L B T M T X N G D C F NAME    TIO T      EDIT      CHR CHR
--------------------------------------------------------------------------------------------------------------------------------
(R1)     ( 0)  (CUSTNO  ) 001-1          (1)                                                                 (ASLGEN'D)     25
(R1)           (ORDERNO ) 002-2          (2)                                                                 (ASLGEN'D)     26
(R1)           (ORDRDATE) 002-2          (2)                                                                 (ASLGEN'D)     27
(R1)           (ORPERSON) 002-2          (2)                                                                 (ASLGEN'D)     28
(R1)     ( 1)  (ORDPONUM) 002-2          (2)                                                                 (ASLGEN'D)     29
(R1)     ( 1)  (ORDDUDAT) 002-2          (2)                                                                 (ASLGEN'D)     30
(R1)     ( 1)  (ORDINVGN) 002-2          (2)                                                                 (ASLGEN'D)     31
(R1)     ( 1)  (ORDCMPLT) 002-2          (2)                                                                 (ASLGEN'D)     32
(R1)     ( 1)  (ITEMORD ) 005-3                                                                              (ASLGEN'D)     33
(R1)     ( 1)  (ITEMNAME) 005-3                                                                              (ASLGEN'D)     34
(R1)     ( 1)  (ITMPRICE) 005-3                                                                              (ASLGEN'D)     35
--------------------------------------------------------------------------------------------------------------------------------
STMT  SEQ                                                                                                      SEQUENCE   LIST
TYPE  NO.                                          TEXT                                                        73 TO 80    SEQ
--------------------------------------------------------------------------------------------------------------------------------
(T1)  (101)              (BUILD THE CUSTOMER MASTER FILE FOR THE ADVANCED TRANSACTION)                          (ASLGEN'D)     36
(T1)  (102)              (S EXAMPLES#                                                 )                         (ASLGEN'D)     37
(T1)  (103)              (LIST THE INITIALLY CREATED CUSTOMER MASTER#                  )                         (ASLGEN'D)     38
FEB 11, 2002  14.27.14                                                                                                  PAGE      4
** MK4W204  TYPE 0  NUMBER OF MESSAGES PRINTED IS    0.                                                                        *****
                    M4NEW    -       15 RECORDS OUTPUT
                    M4TRAN   -       59 RECORDS INPUT
                    M4REPO   -       48 RECORDS OUTPUT
                    M4INPUT  -       18 RECORDS INPUT
                    M4LIST   -      141 RECORDS OUTPUT
                      15 TRACKS ASSIGNED TO M4LIB    --   13 TRACKS NOT FULL. LIBRARY DIRECTORY BLOCKING FACTOR IS  800.
** MK4W209  TYPE 0  1011944 BYTES OF MAIN STORAGE UNUSED DURING DECODING PHASE                                          *****
                    1012638 BYTES OF MAIN STORAGE UNUSED DURING COMPILATION PHASE
                     565128 BYTES OF MAIN STORAGE UNUSED DURING PROCESSING PHASE
```

Figure D-75 CUSTMAST Create Run: Application Coding (Page 2 of 2)

## Listing of CUSTMAST Contents

```
** MK4ED02  TYPE 0  REQUEST-REPT   1  REPORT-1  REQUESTOR ID-                                                    *****
                          BUILD THE CUSTOMER MASTER FILE FOR THE ADVANCED TRANSACTIONS EXAMPLES                 PAGE 1
                                        LIST THE INITIALLY CREATED CUSTOMER MASTER
-------------------------------------------------------------------------------------------------------------------
CUSTOMER  ORDER   ORDER   SALES  PURCH. ORDER  DUE    INVOICE     ORDER    STOCK NO.     DESCRIPTION        PRICE
 NUMBER   NUMBER  DATE    PERSON   NUMBER     DATE  GENERATED? COMPLETED? ORDERED
-------------------------------------------------------------------------------------------------------------------
 00001    10001  010401   PEG      A0001    100101     N          N       G38      1" BINDER NOTEBOOK      $5.00
                                                                          G40      V:BLDR REFERENCE GDE     35.00
          10002  080101   BRR      A0002    110101     N          N       M14      V:BLDR ASL REF GDE      $31.00
                                                                          M23      V:TRAN SCREEN GDE        25.00
 00002      -      -       -        -          -        -          -          -                -          $-
 00115    11501  071201   PEG      A1151    073001     Y          Y       010      ACCOUNTS PAYABLE UM     $45.00
          11502  072201   PEG      A1152    083101     Y          N       072      V:BLDR CLASS GUIDE-B    $55.00
                                                                          106      V:BLDR TABLE TIPS         5.00
                                                                          110      V:BLDR REPORT TIPS        5.00
 00131    31135  050501   JAM      K123J    070101     Y          Y       D77      V:TRAN CLASS GDE-A      $97.50
          31735  070701   JAM      K132J    080101     Y          Y       027      V:BLDR CLASS GUIDE-A    $55.00
 00294    29401  042901   JEE      B2941    060101     Y          Y       978      PURCHASE ORDER GDE      $35.00
 00343    34301  071101   ANS      A3431    080301     Y          Y       865      V:INFM REFERENCE GDE    $39.00
                                                                          866      V:INFM INSTALL GDE        19.00
                                                                          873      V:INFM OPS GUIDE          19.00
          34302  080101   PEG      A3423    090501     N          N       852      PURCHASE ORDER ENVR     $24.00
                                                                          872      G/L REFERENCE GUIDE       75.00
 00551    51135  060601   JAM      K123J    070101     Y          Y       D77      V:TRAN CLASS GDE-A      $97.50
          51755  070901   JAM      K132J    080901     Y          Y       027      V:BLDR CLASS GUIDE-A    $55.00
 00731      -      -       -        -          -        -          -          -                -          $-
 00738    73891  011401   JEF      B7381    012901     Y          Y       L06      V:INFM CLASS PACK      $299.99
                                                                          L06F     V:INFM CLASS SLIDES       99.00
                                                                          L96      V:BLDR APPL GUIDE         50.00
 00818      -      -       -        -          -        -          -          -                -          $-
 00847      -      -       -        -          -        -          -          -                -          $-
 00867    86731  061501   WAV      W1531    100101     Y          Y       013      V:BLDR REF SUMMARY      $51.00
          86751  071501   SUR      S1552    110901     Y          Y       041      V:BLDR PROCEDURE GDE    $55.00
 00918    91801  051201   GEE      A9181    071501     Y          Y       L97      V:INFM QUERY GUIDE      $45.00
                                                                          M01      V:TRAN ENV GDE CICS       47.50
 00951    91395  090601   JOH      9123A    010902     N          N       013      V:BLDR REF SUMMARY      $51.00
          91795  090901   UNC      9132B    010902     N          N       041      V:BLDR PROCEDURE GDE    $55.00
 01000      -      -       -        -          -        -          -          -                -          $-
** MK4FT06  TYPE 1        50 FIELDS WERE PRINTED AS INVALID, MISSING OR OVERFLOW.                              *****
** MK4FT03  TYPE 0  END OF REPORT.                                                                            *****
** MK4W204  TYPE 0  NUMBER OF MESSAGES PRINTED IS    3.                                                      *****
                    M4REPO   -       47 RECORDS INPUT
                    M4LIST   -       43 RECORDS OUTPUT
** MK4W209  TYPE 0   999768 BYTES OF MAIN STORAGE UNUSED DURING REPORTING PHASE                               *****
```

Figure D-76 CUSTMAST Create Run: Listing of CUSTMAST Contents

# Updating Application

## Banner Page

```
FEB 20, 2002  16.03.15                                                                              PAGE      1
                                        ******************************************
                                        * VISION:Builder   4400  (OS/390 - 14.0)  *
                                        *          COPYRIGHT (C) 2002            *
                                        * COMPUTER ASSOCIATES INTERNATIONAL, INC. *
                                        ******************************************
VISION:Builder JCR1 14.0 021402              ENABLED FOR IBM LANGUAGE ENVIRONMENT              BUILD STAMP = 102045,12:39:11.
                                          COMLIB  MODIFICATIONS (PTFs, APARs) INSTALLED
                                    201,202,203,204,205,206,207,208,209,210,212,213,214,215,216
=====================================================================================================================
=                                                                                                                   =
=                  I N S T A L L A T I O N   P A R A M E T E R S   ( M 4 P A R A M S , M A R K L I B P )            =
=                                                                                                                   =
=  SYSTEM DELIMITER:   #          PAGE HEIGHT:       66         M4LIST WIDTH:      132        DEF WIDTH OF PAGE:  0        =
=                                                                                                                   =
=  AUTO GRAND:         N          HEADING CHAR:      -          SUBTITLE REPEAT:   N          INVALID FIELD:      *        =
=                                                                                                                   =
=  MISSING FIELD:      -          NON-EDIT FIELD:    +          PERCENT CHAR:      %          LEFT SEPARATOR:     (        =
=                                                                                                                   =
=  RIGHT SEPARATOR:    )          SINGLE SEPARATOR:  ,          SOURCE SPACING:    1          PRINT MESSAGES:     Y        =
=                                                                                                                   =
=  CONSOLE MESSAGES:   N          M4REPO BLOCKSIZE:  4,096      INPUT I/O BUFFERS: 2          OUTPUT I/O BUFFERS: 1        =
=                                                                                                                   =
=  SNGL-STEP STORAGE:  8,192      SNGL-STEP SORTSIZE: 524,288   DIGIT SELECT CHAR: 9          ZERO SUPPRESS CHAR: Z        =
```

Figure D-77 Updating Application: Banner Page (Page 1 of 2)

```
=                                                                                                                                  =
= CURRENCY CHAR:      $          PLUS CHAR:          +          MINUS CHAR:         -          CHECK PROTECT CHAR: *          =
=                                                                                                                                  =
= DECIMAL CHAR:       .          GROUPING CHAR:      ,          PRIMARY PLOT CHAR:  X          SECONDARY PLOT CHAR:*          =
=                                                                                                                                  =
= FIT PLOT CHAR:      .          HORIZONTAL AXIS:    _          HORIZONTAL HASH:    |          VERTICAL AXIS:      |          =
=                                                                                                                                  =
= VERTICAL HASH:      -          MINUTES/HOUR:       60         SECONDS/MINUTE:     60         TIME DELIMITER:     HH:MM:SS   =
=                                                                                                                                  =
= DATE FORMAT:      MMM DD, YYYY TODAY FORMAT+DELIM: MM/DD/YY   ISDATE DELIMITER:   YYYY-MM-DD JULIAN DELIMITER:   YY.DDD     =
=                                                                                                                                  =
= SORT PROGRAM CODE:  2          MINCORE VALUE:      12 K       ALT M4LIST WIDTH:   264        ALT DEF W/OF PAGE:  0          =
=                                                                                                                                  =
= MAX LINES OF TRACE: 1,024      ITEM TRACKING:      0          SUPPRESS NDS REPT?: N          DEFAULT MAXGETMN:   1,024 K    =
=                                                                                                                                  =
= CONDITION CODE 1:   0          CONDITION CODE 2:   4          CONDITION CODE 3:   8          CONDITION CODE 4:   16         =
=                                                                                                                                  =
= ISAM INDEX INCORE:  N          M4LIB BLKG FACTOR:  0          M4LIB RESERVE:      0          M4LIB COMPONENT:    C4.5 B/V   =
=                                                                                                                                  =
= DEFAULT F/P AMODE:  24         M4LIB COMPRESSION?: N          M4LIB COMPRESS MIN: 507                                        =
=                                                                                                                                  =
==================================================================================================================================
```

Figure D-77 Updating Application: Banner Page (Page 2 of 2)

## Run Control Commands

```
FEB 20, 2002  16.03.15                                                                                                  PAGE      2
----------------------------------------------------------------------------------------------------------------------------------
  RUN     STMT    FILE    O N S D U T R A B S D S L  RB  R S M R S            START           END              SEQUENCE   LIST
  NAME    TYPE    NAME    L E E I P R E U U R L C S  PL  J O O F S            SEARCH          SEARCH           73 TO 80    SEQ
                          D W Q R D N P D F T M N T  TK  T P P O R
----------------------------------------------------------------------------------------------------------------------------------
(UPDCUSTM) (RC) (CUSTMAST) (S,S)      (S,S)       (T,G)          (R)                                                (ASLGEN'D)     1
    CONTROL NAME UPDCUSTM TERM LISTGEN                                                                                             2
    FILE MASTER INPUT  CUSTMAST                                                                                                    3
    FILE MASTER OUTPUT                                                                                                             4
    FILE TRAN   TRANWORK GROUPS ORDENTRY                                                                                           5
    FILE CORD1  ITEMMAST DIRECT BY X.ITM5ORD1                                                                                      6
    FILE REPORT                                                                                                                    7
    ;                                                                                                                              8
----------------------------------------------------------------------------------------------------------------------------------
  RUN     STMT  LOGICAL  LBL  DTF/DD      KEY       KEY       KEY      QUALIFIER   USER I/O  ARRAY      PASSWORD/  SEQUENCE   LIST
  NAME    TYPE  FILENAME       NAME     FIELD-1   FIELD-2   FIELD-3   ICF CRD DBI   MODULE   QUAL       AUTH.-ID   73 TO 80    SEQ
----------------------------------------------------------------------------------------------------------------------------------
(UPDCUSTM) (RF) (CUSTMAST) (S) (M4OLD   )                                                                          (ASLGEN'D)     9
(UPDCUSTM) (RF) (CUSTMAST) (S) (M4NEW   )                                                                          (ASLGEN'D)    10
(UPDCUSTM) (RF) (ITEMMAST) (S) (M4CORD1 ) (ITM5ORD1)                    (X)                                       (ASLGEN'D)    11
(UPDCUSTM) (RF) (TRANWORK) (S) (M4TRAN  )                                                                          (ASLGEN'D)    12
----------------------------------------------------------------------------------------------------------------------------------
  RUN     STMT                  TRANSACTION GROUPS                                                                  SEQUENCE   LIST
  NAME    TYPE     1         2         3         4         5                                                        73 TO 80    SEQ
----------------------------------------------------------------------------------------------------------------------------------
(UPDCUSTM) (RG) (ORDENTRY)                                                                                         (ASLGEN'D)    13

==================================================================================================================================
=                                                                                                                                  =
=                                                  F I L E   S U M M A R Y                                                         =
=                                                                                                                                  =
= JOBNAME: KREJO04X                          STEPNAME: PROC1                                                                       =
=                                                                                                                                  =
=  FILE NAME: M4OLD                          DSNAME: BUILDER.R140.CUSTMAST.BUILT.R15#1                                             =
=  FILE NAME: M4NEW                          DSNAME: BUILDER.R140.CUSTMAST.UPDATED.R15#1                                           =
=  FILE NAME: M4TRAN                         DSNAME: KREJO04.KREJO04X.JOB03151.D0000101.?                                          =
=  FILE NAME: M4CORD1                        DSNAME: BUILDER.R140.ITEM.MAST.R15#1                                                  =
=  FILE NAME: M4LIB                          DSNAME: BUILDER.R140.M4LIB.R15#1                                                      =
=                                                                                                                                  =
==================================================================================================================================
```

Figure D-78 Updating Application: Run Control Commands

## Type 1 Procedure TRANPREP

```
FEB 20, 2002  16.03.15                                                                                          PAGE     3
                                                    *****************************
                                                    *  PROC NAME - TRANPREP     *
                                                    *  INPUT STREAM PROCEDURE   *
                                                    *****************************
--------------------------------------------------------------------------------------------------------------------------
STMT REPORT                                  MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ        BACK BR RNT AB      SEQUENCE  LIST
TYPE  DATE         REQUESTOR ID             ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM     73 TO 80   SEQ
--------------------------------------------------------------------------------------------------------------------------
(ER)                                                                          (1)                              (ASLGEN'D)    14
     TRANPREP: PROC TYPE1                                                                                                    15
     ;                                                                                                                       16
     ; This TYPE 1 Procedure is executed for each new transaction record                                                     17
     ;                                                                                                                       18
     ;   Either Read from the M4TRAN Input file or                                                                           19
     ;   or because of a RETRY, a TRANSFER TYPE_1 Command                                                                    20
     ;                                                                                                                       21
     ITMCOUNT: FIELD P 1  HEADING 'COUNT' 'ITEM TRANS'                                                                       22
     WORK1C:   FIELD C 1  HEADING 'WORK' 'CHAR 1'                                                                            23
     ERRNOTE:  FIELD C 25 HEADING 'ERROR' 'COMMENT'                                                                          24
     RETRYSW:  FIELD C 3  HEADING 'RETRY' 'SWITCH' INIT 'NO '                                                                25
     ;                                                                                                                       26
     LET T.ERRNOTE = ' '                                                                                                     27
     IF T.RETRYSW = 'NO'              ; RETRY ? Entry                                                                        28
       REPORT F.M4TRAN X.ALL80        ; No, list Tran Rcd                                                                    29
       TITLE 'LISTING OF TRANSACTION RECORDS 80/80 WITH RECORD COUNT'                                                        30
       END REPORT                                                                                                            31
     END IF                                                                                                                  32
     IF X.TRANID = '3ADDORDR'         ; if a 3ADDORDR Tran Rcd                                                               33
       LET T.ITMCOUNT = 0             ; reset the ITEM COUNTER                                                               34
     END IF                                                                                                                  35
     IF X.TRANID = '5ADDITEM'         ; if a 5ADDITEM Tran Rcd                                                               36
       LET T.ITMCOUNT = T.ITMCOUNT + 1  ; Add to the ITEM COUNTER                                                            37
       IF X.ITM5QTY1 LT 1             ; Is there a Quantity                                                                  38
        LET F.DELETE = 4              ; No, reject the transaction                                                           39
        LET T.WORK1C = T.ITMCOUNT     ; Convert item count                                                                   40
        COMBINE 'QTY INVALID FROM SET' T.WORK1C STORE T.ERRNOTE BLANKS 1                                                     41
       END IF                                                                                                                42
     END IF                                                                                                                  43
     IF X.TRANID NE '3ADDORDR' '5ADDITEM'     ;Expected IDs?                                                                 44
        LET F.DELETE = 4                      ; No, reject the tran rcd                                                      45
       END IF                                                                                                                46
     ;                                                                                                                       47
     END PROC ;TRANPREP                                                                                                      48
     ;                                                                                                                       49
--------------------------------------------------------------------------------------------------------------------------
STMT   FIELD    FIELD FLD DEC OUTPT EDIT     INITIAL         LINE 1 OF       LINE 2 OF                         SEQUENCE  LIST
TYPE   NAME     LNGTH TYP PLC EDIT  LGTH      VALUE        COLUMN HEADING  COLUMN HEADING                      73 TO 80   SEQ
--------------------------------------------------------------------------------------------------------------------------
(TF) (ITMCOUNT) (  1) (P)                                  (COUNT        ) (ITEM TRANS     )                   (ASLGEN'D)    50
(TF) (WORK1C  ) (  1) (C)                                  (WORK         ) (CHAR 1         )                   (ASLGEN'D)    51
(TF) (ERRNOTE ) ( 25) (C)                                  (ERROR        ) (COMMENT        )                   (ASLGEN'D)    52
(TF) (RETRYSW ) (  3) (C)                 (NO            ) (RETRY        ) (SWITCH         )                   (ASLGEN'D)    53
(TF) (TEMP__4)  ( 20) (C)                 (QTY INVALID FROM)                                                   (ASLGEN'D)    54
FEB 20, 2002  16.03.15                                                                                          PAGE     4
--------------------------------------------------------------------------------------------------------------------------
STMT SEQ  LOG CON .....OPERAND-A.....  OPER  ............OPERAND-B.............  ......RESULT......  PARTIAL FIELD  SEQUENCE  LIST
TYPE NO.  LEV CTR QLF FIELD   SEG-LVL  ATN   QLF FIELD   SEG-LVL  (OR CONSTANT)  QLF FIELD  SEG-LVL  STRT NUMB OPR  73 TO 80   SEQ
--------------------------------------------------------------------------------------------------------------------------
(PR) (100)                             (R )  (C)                                 (T,ERRNOTE )                      (ASLGEN'D)    55
(PR) (101)        (T,RETRYSW )         (EQ)  (C,NO                             )                                   (ASLGEN'D)    56
(PR) (102)                             (NS)    (104                            )                                   (ASLGEN'D)    57
(PR) (103)                             (GO)    (SUB REPT__2                    )                                   (ASLGEN'D)    58
(PR) (104)        (X,TRANID  ) 001-1   (EQ)  (C,3ADDORDR                       )                                   (ASLGEN'D)    59
(PR) (105)                             (NS)    (107                            )                                   (ASLGEN'D)    60
(PR) (106)                             (R )  (D,0                              ) (T,ITMCOUNT)                      (ASLGEN'D)    61
(PR) (107)        (X,TRANID  ) 001-1   (EQ)  (C,5ADDITEM                       )                                   (ASLGEN'D)    62
(PR) (108)                             (NS)    (116                            )                                   (ASLGEN'D)    63
(PR) (109)        (T,ITMCOUNT)         (+ )  (D,1                              ) (T,ITMCOUNT)                      (ASLGEN'D)    64
(PR) (110)        (X,ITM5QTY1) 001-1   (LT)  (D,1                              )                                   (ASLGEN'D)    65
(PR) (111)                             (NS)    (116                            )                                   (ASLGEN'D)    66
(PR) (112)                             (R )  (D,4                              ) (F,DELETE  )                      (ASLGEN'D)    67
(PR) (113)                             (R )  (T,ITMCOUNT)                        (T,WORK1C  )                      (ASLGEN'D)    68
(PR) (114)                             (R )  (C,QTY INVALID FROM SET           ) (T,TEMP__4)                       (ASLGEN'D)    69
(PR) (115)        (T,TEMP__4)          (C1)  (T,WORK1C  )                        (T,ERRNOTE )                      (ASLGEN'D)    70
(PR) (116)        (X,TRANID  ) 001-1   (NE)  (C,3ADDORDR                       )                                   (ASLGEN'D)    71
(PR) (117)                             (NS)    (END                            )                                   (ASLGEN'D)    72
(PR) (118)        (X,TRANID  ) 001-1   (NE)  (C,5ADDITEM                       )                                   (ASLGEN'D)    73
(PR) (119)                             (NS)    (END                            )                                   (ASLGEN'D)    74
(PR) (120)                             (R )  (D,4                              ) (F,DELETE  )                      (ASLGEN'D)    75
FEB 20, 2002  16.03.15                                                                                          PAGE     5
                                                    *****************************
```

Figure D-79  Updating Application: Type 1 TRANPREP (Page 1 of 2)

```
                                        *  REQUEST NAME - REPT__2  *
                                        *  GENERATED REQUEST ____  *
                                        ****************************

-------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                              MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID        ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM      73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(ER)                                                                          (S)                              (ASLGEN'D)     76
-------------------------------------------------------------------------------------------------------------------------------
 ST SUM V  8  PG  PG   I SP  MAXIMUM TB COLUMN  DT  PG  STRT LIN SL RH .....SUBFILE..... C E A S R S N      C  SEQUENCE   LIST
 TY RPT S LPI WID HGT  M FRM LPP PGS TO HEADING POS POS PAGE NUM UB PN ENT    NAME   BLK/ FMT H F G T S U D      O  73 TO 80    SEQ
 MP     P             G            LT TYP POS         NUM      ML TD REC          BUF      C C S L T M S      N
-------------------------------------------------------------------------------------------------------------------------------
(E1)                                                                                                           (ASLGEN'D)     77
-------------------------------------------------------------------------------------------------------------------------------
         SP  Q              E N S D C S T C C M M A M D Q PCT/RAT     P              PARTIAL   SEQUENCE   LIST
STMT SEQ BF  L  FIELD  SEG- N P E E T U O U N A I V O E L  FIELD  RA- C     PICTURE     STRT NO  73 TO 80    SEQ
TYPE NO. COL F  NAME     LVL D R Q S L B T M T X N G D C F  NAME   TIO T      EDIT      CHR CHR
-------------------------------------------------------------------------------------------------------------------------------
(R1)         (F,M4TRAN  )                                                                                      (ASLGEN'D)     78
(R1)         (X,ALL80   ) 001-1                                                                                (ASLGEN'D)     79
-------------------------------------------------------------------------------------------------------------------------------
STMT  SEQ                                                                                                      SEQUENCE   LIST
TYPE  NO.                                          TEXT                                                        73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(T1) (101)                 (LISTING OF TRANSACTION RECORDS 80/80 WITH RECORD COUNT#    )                       (ASLGEN'D)     80
```

Figure D-79 Updating Application: Type 1 TRANPREP (Page 2 of 2)

## Type 2 Procedure ORDERCHK

```
FEB 20, 2002  16.03.15                                                                                    PAGE      6

                                        ****************************
                                        *  PROC NAME - ORDERCHK    *
                                        *  INPUT STREAM PROCEDURE  *
                                        ****************************

-------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                              MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID        ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM      73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(ER)                                                                          (2)                              (ASLGEN'D)     81
    ORDERCHK: PROC TYPE2                                                                                                      82
    ;                                                                                                                         83
    ; This TYPE 2 Procedure is executed for each new transaction record                                                       84
    ;  not "automatically' rejected based on the Transaction Definition                                                       85
    ;                                                                                                                        86
    ; We are aligned with the Customer Master Record, but the update from                                                     87
    ; this Tran Rcd has not been applied yet.                                                                                 88
    ;                                                                                                                        89
    ; For the 3ADDORDR Transaction Records                                                                                   90
    ;  we will check the Tran Rcd "order date" to see if it pre-dates                                                         91
    ;  any other orders in the customer record to provide a information                                                       92
    ;  listing for the order desk                                                                                            93
    ;                                                                                                                        94
    MSTRDATE: FIELD D 4 HEADING 'MASTER' 'DATE' ; Lilian Date                                                               95
    TRANDATE: FIELD D 4 HEADING 'TRAN'   'DATE' ; Lilian Date                                                               96
    ;                                                                                                                       97
    IF X.TRANID = '3ADDORDR'                                                                                                98
      LET T.TRANDATE = X.ORDRDATE        ; convert date to Lilian                                                           99
      LET T.MSTRDATE = N.ORDRDATE        ; convert date to Lilian                                                          100
      IF  T.TRANDATE < T.MSTRDATE                                                                                          101
       REPORT X.TRANID X.CUSTNO X.ORDERNO T.TRANDATE T.MSTRDATE F.M4TRAN                                                   102
       TITLE LINE 'LIST OF ORDERS WITH ORDER DATES EARLIER THAN MASTER',                                                   103
                  ' FILE ORDER DATES'                                                                                      104
       END REPORT                                                                                                         105
      END IF                                                                                                              106
    END IF                                                                                                                107
    END PROC ;ORDERCHK                                                                                                    108
    ;                                                                                                                     109
-------------------------------------------------------------------------------------------------------------------------------
STMT   FIELD    FIELD FLD DEC OUTPT EDIT     INITIAL        LINE 1 OF       LINE 2 OF                          SEQUENCE   LIST
TYPE   NAME     LNGTH TYP PLC EDIT  LGTH      VALUE        COLUMN HEADING  COLUMN HEADING                      73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(TF) (MSTRDATE) (  4) (D)                                 (MASTER        ) (DATE          )                    (ASLGEN'D)    110
(TF) (TRANDATE) (  4) (D)                                 (TRAN          ) (DATE          )                    (ASLGEN'D)    111
-------------------------------------------------------------------------------------------------------------------------------
STMT SEQ  LOG CON .....OPERAND-A.....  OPER .............OPERAND-B.............   ......RESULT......  PARTIAL FIELD  SEQUENCE   LIST
TYPE NO.  LEV CTR QLF FIELD   SEG-LVL  ATN  QLF FIELD   SEG-LVL  (OR CONSTANT)   QLF FIELD  SEG-LVL  STRT NUMB OPR  73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(PR) (100)         (X,TRANID  ) 001-1  (EQ)   (C,3ADDORDR                    )                                    (ASLGEN'D)    112
(PR) (101)                             (NS)     (END                         )                                    (ASLGEN'D)    113
(PR) (102)                             (R )   (X,ORDRDATE) 001-1               (T,TRANDATE)                   (ASLGEN'D)    114
```

Figure D-80 Updating Application: Type 2 Procedure ORDERCHK (Page 1 of 2)

```
(PR) (103)                          (R )  (N,ORDRDATE) 002-2                (T,MSTRDATE)                        (ASLGEN'D)   115
(PR) (104)          (T,TRANDATE)    (LT)  (T,MSTRDATE)                                                              (ASLGEN'D)   116
(PR) (105)                          (NS)    (END                          )                                         (ASLGEN'D)   117
(PR) (106)                          (GO)    (SUB REPT___5                 )                                         (ASLGEN'D)   118
FEB 20, 2002  16.03.15                                                                                                   PAGE     7

                                                  ****************************
                                                  *  REQUEST NAME - REPT___5  *
                                                  *  GENERATED REQUEST        *
                                                  ****************************

-------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                              MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID        ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM      73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(ER)                                                                                (S)                              (ASLGEN'D)   119
-------------------------------------------------------------------------------------------------------------------------------
 ST SUM V  8  PG  PG   I SP  MAXIMUM TB COLUMN  DT  PG  STRT LIN SL RH .......SUBFILE..... C E A S R S N      C  SEQUENCE   LIST
 TY RPT S LPI WID HGT  M FRM LPP PGS TO HEADING POS POS PAGE NUM UB PN ENT    NAME   BLK/ FMT H F G T S U D      O  73 TO 80    SEQ
 MP     P            G            LT TYP POS         NUM      ML TD REC           BUF      C C S L T M S      N
-------------------------------------------------------------------------------------------------------------------------------
(E1)                                                                                                                 (ASLGEN'D)   120
-------------------------------------------------------------------------------------------------------------------------------
         SP  Q               E  N  S  D  C  S  T  C  C  M  M  A  M  D Q PCT/RAT     P                PARTIAL   SEQUENCE   LIST
STMT SEQ BF  L  FIELD   SEG-  N  P  E  E  T  U  O  U  N  A  I  V  O  E L  FIELD   RA- C     PICTURE    STRT NO  73 TO 80    SEQ
TYPE NO. COL F  NAME      LVL D  R  Q  S  L  B  T  M  T  X  N  G  D  C F  NAME    TIO T      EDIT      CHR CHR
-------------------------------------------------------------------------------------------------------------------------------
(R1)         (X,TRANID  ) 001-1                                                                                      (ASLGEN'D)   121
(R1)         (X,CUSTNO  ) 001-1                                                                                      (ASLGEN'D)   122
(R1)         (X,ORDERNO ) 001-1                                                                                      (ASLGEN'D)   123
(R1)         (T,TRANDATE)                                                                                            (ASLGEN'D)   124
(R1)         (T,MSTRDATE)                                                                                            (ASLGEN'D)   125
(R1)         (F,M4TRAN  )                                                                                            (ASLGEN'D)   126
-------------------------------------------------------------------------------------------------------------------------------
STMT  SEQ                                                                                                            SEQUENCE   LIST
TYPE  NO.                                          TEXT                                                              73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(T1) (101)              (LIST OF ORDERS WITH ORDER DATES EARILER THAN MASTER FILE OR)                                 (ASLGEN'D)   127
(T1) (102)              (DER DATES#                                                 )                                (ASLGEN'D)   128
```

Figure D-80 Updating Application: Type 2 Procedure ORDERCHK (Page 2 of 2)

## Type M Procedure UPDATRPT

```
FEB 20, 2002  16.03.15                                                                                                   PAGE     8

                                                  ****************************
                                                  *  PROC NAME - UPDATRPT     *
                                                  *  INPUT STREAM PROCEDURE   *
                                                  ****************************

-------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                              MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID        ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM      73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(ER)                                                                                (M)                              (ASLGEN'D)   129
    UPDATRPT: PROC TYPEM                                                                                                      130
    ;                                                                                                                         131
    ; This TYPE M Procedure executes after each Transaction Record                                                            132
    ; Updates the Customer Record.                                                                                           133
    ;                                                                                                                         134
    ; References to Segments just updated by the current Transaction Record                                                   135
    ; will not cause Looping on those Segment Types.                                                                         136
    ; If the current transaction record is an 5ADDITEM,                                                                      137
    ;  We need to get the ITEM NAME and PRICE from the ITEM File into                                                         138
    ;  the Item Segment of the Customer Master.                                                                               139
    ;                                                                                                                         140
    ;  A Direct Read of the ITEM File (CORD1) using the ITEM Order Id                                                         141
    ;  If we do not find an Item Record, DELETE the segment from the                                                          142
    ;  Customer Record and report the error                                                                                   143
    ;                                                                                                                         144
    IF X.TRANID = '5ADDITEM'                ; Item Trans                                                                    145
     IF 1.ITEMNO = 1.ITEMNO                 ; Yes, This READS Item File                                                     146
      LET N.ITEMNAME = 1.ITEMNAME           ; RCD Found, Get data                                                           147
      LET N.ITMPRICE = 1.ITMPRICE                                                                                           148
     ELSE                                   ; No RCD Found                                                                  149
      LET T.ERRNOTE = 'ITEM ORDERED NOT ON FILE'                                                                            150
      CALL REPORT REJRPTS                   ; List error                                                                    151
      LET N.ITEMORD = ' '                   ; Clear Segment Key                                                             152
      LET F.DELETE = 2                      ; Mark SEG for DELETE                                                           153
      GO TO ENDPROCM                        ; DO NOT REPORT                                                                 154
     END IF                                                                                                                 155
    END IF                                                                                                                  156
    REPORT CUSTNO ORDERNO ORDRDATE ORPERSON ORDPONUM ORDDUDAT,                                                              157
           ORDINVGN ORDCMPLT ITEMORD ITEMNAME ITMPRICE F.STRAN                                                              158
      GROUP BY CUSTNO AT LEVEL 1                                                                                            159
```

Figure D-81 Updating Application: Type M Procedure UPDATRPT (Page 1 of 2)

```
      GROUP BY ORDERNO ORDRDATE ORPERSON ORDPONUM,                                                                    160
            ORDDUDAT ORDINVGN ORDCMPLT AT LEVEL 2                                                                    161
      ITEM CUSTNO,                                                                                                   162
           SPACES 0                                                                                                  163
      ITEM ORDPONUM ORDDUDAT ORDINVGN ORDCMPLT,                                                                      164
           ITEMORD ITEMNAME ITMPRICE,                                                                                165
           SPACES 1                                                                                                  166
      TITLE 'THE UPDATED CUSTOMER MASTER FILE RECORDS'                                                               167
    END REPORT                                                                                                       168
    ;                                                                                                                169
    ENDPROCM: END PROC ;UPDATRPT                                                                                     170
    ;                                                                                                                171
-------------------------------------------------------------------------------------------------------------------------
STMT SEQ  LOG CON .....OPERAND-A.....  OPER  .............OPERAND-B.............   ......RESULT......  PARTIAL FIELD  SEQUENCE   LIST
TYPE NO.  LEV CTR QLF FIELD   SEG-LVL  ATN   QLF FIELD   SEG-LVL  (OR CONSTANT)   QLF FIELD  SEG-LVL  STRT NUMB OPR  73 TO 80   SEQ
-------------------------------------------------------------------------------------------------------------------------
(PR) (100)         (X,TRANID  ) 001-1  (EQ)    (C,5ADDITEM                    )                                 (ASLGEN'D)   172
(PR) (101)                             (NS)      (112                         )                                 (ASLGEN'D)   173
FEB 20, 2002  16.03.15                                                                                          PAGE       9
-------------------------------------------------------------------------------------------------------------------------
STMT SEQ  LOG CON .....OPERAND-A.....  OPER  .............OPERAND-B.............   ......RESULT......  PARTIAL FIELD  SEQUENCE   LIST
TYPE NO.  LEV CTR QLF FIELD   SEG-LVL  ATN   QLF FIELD   SEG-LVL  (OR CONSTANT)   QLF FIELD  SEG-LVL  STRT NUMB OPR  73 TO 80   SEQ
-------------------------------------------------------------------------------------------------------------------------
(PR) (102)         (1,ITEMNO  ) 001-1  (EQ)    (1,ITEMNO  ) 001-1                                                (ASLGEN'D)   174
(PR) (103)                             (NS)      (107                         )                                 (ASLGEN'D)   175
(PR) (104)                             (R )    (1,ITEMNAME) 001-1               (N,ITEMNAME) 005-3              (ASLGEN'D)   176
(PR) (105)                             (R )    (1,ITMPRICE) 001-1               (N,ITMPRICE) 005-3              (ASLGEN'D)   177
(PR) (106)                             (GO)      (112                         )                                 (ASLGEN'D)   178
(PR) (107)                             (R )    (C,ITEM ORDERED NOT ON FILE    ) (T,ERRNOTE )                  (ASLGEN'D)   179
(PR) (108)                             (GO)      (SUB REJRPTS                 )                                 (ASLGEN'D)   180
(PR) (109)                             (R )    (C)                              (N,ITEMORD ) 005-3              (ASLGEN'D)   181
(PR) (110)                             (R )    (D,2                           ) (F,DELETE  )                    (ASLGEN'D)   182
(PR) (111)                             (GO)      (END                         )                                 (ASLGEN'D)   183
(PR) (112)                             (GO)      (SUB REPT___6                )                                 (ASLGEN'D)   184
FEB 20, 2002  16.03.15                                                                                          PAGE      10

                                          ****************************
                                          *  REQUEST NAME - REPT___6 *
                                          *  GENERATED REQUEST       *
                                          ****************************

-------------------------------------------------------------------------------------------------------------------------
STMT REPORT                          MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB      SEQUENCE   LIST
TYPE  DATE        REQUESTOR ID       ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM      73 TO 80   SEQ
-------------------------------------------------------------------------------------------------------------------------
(ER)                                                                          (S)                               (ASLGEN'D)   185
-------------------------------------------------------------------------------------------------------------------------
 ST SUM V  8  PG  PG   I SP  MAXIMUM TB COLUMN  DT  PG  STRT LIN SL RH .......SUBFILE........ C E A S R S N      C  SEQUENCE   LIST
 TY RPT S LPI WID HGT  M FRM LPP PGS TO HEADING POS POS PAGE NUM UB PN ENT    NAME   BLK/ FMT H F G T S U D      O  73 TO 80   SEQ
 MP     P            G            LT TYP POS         NUM      ML TD REC          BUF      C C S L T M S      N
-------------------------------------------------------------------------------------------------------------------------
(E1)                                                                                                            (ASLGEN'D)   186
-------------------------------------------------------------------------------------------------------------------------
         SP  Q              E N S D C S T C C M M A M D Q PCT/RAT     P              PARTIAL   SEQUENCE   LIST
STMT SEQ BF  L  FIELD  SEG- N P E E T U O U N A I V O E L  FIELD  RA- C    PICTURE   STRT NO  73 TO 80   SEQ
TYPE NO. COL F  NAME     LVL D R Q S L B T M T X N G D C F  NAME   TIO T     EDIT    CHR CHR
-------------------------------------------------------------------------------------------------------------------------
(R1)     ( 0)  (CUSTNO  ) 001-1          (1)                                                                    (ASLGEN'D)   187
(R1)           (ORDERNO ) 002-2          (2)                                                                    (ASLGEN'D)   188
(R1)           (ORDRDATE) 002-2          (2)                                                                    (ASLGEN'D)   189
(R1)           (ORPERSON) 002-2          (2)                                                                    (ASLGEN'D)   190
(R1)     ( 1)  (ORDPONUM) 002-2          (2)                                                                    (ASLGEN'D)   191
(R1)     ( 1)  (ORDDUDAT) 002-2          (2)                                                                    (ASLGEN'D)   192
(R1)     ( 1)  (ORDINVGN) 002-2          (2)                                                                    (ASLGEN'D)   193
(R1)     ( 1)  (ORDCMPLT) 002-2          (2)                                                                    (ASLGEN'D)   194
(R1)     ( 1)  (ITEMORD ) 005-3                                                                                 (ASLGEN'D)   195
(R1)     ( 1)  (ITEMNAME) 005-3                                                                                 (ASLGEN'D)   196
(R1)     ( 1)  (ITMPRICE) 005-3                                                                                 (ASLGEN'D)   197
(R1)         (F,STRAN   )                                                                                       (ASLGEN'D)   198
-------------------------------------------------------------------------------------------------------------------------
STMT  SEQ                                                                                                       SEQUENCE   LIST
TYPE  NO.                                    TEXT                                                               73 TO 80   SEQ
-------------------------------------------------------------------------------------------------------------------------
(T1) (101)          (THE UPDATED CUSTOMER MASTER FILE RECORDS#                )                                 (ASLGEN'D)   199
```

Figure D-81 Updating Application: Type M Procedure UPDATRPT (Page 2 of 2)

## Type 3 Procedure ITEMLOOP

```
FEB 20, 2002  16.03.15                                                                                                PAGE     11
                                                  ****************************
                                                  *  PROC NAME - ITEMLOOP    *
                                                  *  INPUT STREAM PROCEDURE  *
                                                  ****************************

-------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                              MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID        ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM      73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(ER)                                                                            (3)          (    5)             (ASLGEN'D)  200
     ITEMLOOP: PROC TYPE3 BACKBRANCH 5                                                                                        201
     ;                                                                                                                        202
     ; Executed after a Transaction Record has updated the Customer Master                                                   203
     ;  The Entire Master Record (all segment occurrances) is available                                                      204
     ;                                                                                                                        205
     ; We are looking for 5ADDITEM Transaction Records                                                                       206
     ;  For the ITEM Tran Rcds we need to manipulate Item/Qty Sets                                                           207
     ;                                                                                                                        208
     IF X.TRANID = '5ADDITEM'                                                                                                 209
      CALL PROCEDURE ITEMCHKR             ; Go Check Item entries                                                           210
      IF T.RETRYSW = 'YES'                ; Need to RETRY Tran                                                              211
        TRANSFER TYPE_1                   ; Yes, Go back to TYPE 1                                                          212
      END IF                                                                                                                  213
     END IF                                                                                                                   214
     END PROC ;ITEMLOOP                                                                                                       215
     ;                                                                                                                        216
-------------------------------------------------------------------------------------------------------------------------------
STMT SEQ  LOG CON .....OPERAND-A.....  OPER  .............OPERAND-B.............  ......RESULT......  PARTIAL FIELD  SEQUENCE   LIST
TYPE NO.  LEV CTR QLF FIELD   SEG-LVL  AIN   QLF FIELD   SEG-LVL  (OR CONSTANT)   QLF FIELD  SEG-LVL  STRT NUMB OPR  73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(PR) (100)        (X,TRANID  ) 001-1  (EQ)   (C,5ADDITEM                    )                                      (ASLGEN'D)  217
(PR) (101)                            (NS)     (END                         )                                      (ASLGEN'D)  218
(PR) (102)                            (GO)     (SUB ITEMCHKR                )                                      (ASLGEN'D)  219
(PR) (103)        (T,RETRYSW )        (EQ)   (C,YES                         )                                      (ASLGEN'D)  220
(PR) (104)                            (NS)     (END                         )                                      (ASLGEN'D)  221
(PR) (105)                            (GO)     (TYPE 1                      )                                      (ASLGEN'D)  222
```

Figure D-82 Updating Application: Type 3 Procedure ITEMLOOP

## Type 4 Procedure TRANERR

```
FEB 20, 2002  16.03.15                                                                                                PAGE     12
                                                  ****************************
                                                  *  PROC NAME - TRANERR     *
                                                  *  INPUT STREAM PROCEDURE  *
                                                  ****************************

-------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                              MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID        ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM      73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(ER)                                                                            (4)          (    5)             (ASLGEN'D)  223
     TRANERR: PROC TYPE4 BACKBRANCH 5                                                                                         224
     ;                                                                                                                        225
     ; Executed when a Transaction is rejected                                                                               226
     ;  Automatically based on the Transaction Definition Criteria                                                           227
     ;  or Procedurally by the Application Processing Criteria                                                               228
     ;                                                                                                                        229
     IF F.XTRAN = 1                                                                                                           230
       CALL REPORT REJRPTS       ;Report Procedurally Rejected Transactions                                                   231
     ELSE                                                                                                                     232
       LET T.ERRNOTE = 'AUTO REJECT, SEE DIAG MSG'                                                                            233
       CALL REPORT REJRPTS       ;Report Automatically Rejected Transactions                                                  234
     END IF                                                                                                                   235
     ;                                                                                                                        236
     IF X.TRANID = '5ADDITEM'             ; Item ADD Trans,                                                                 237
      CALL PROCEDURE ITEMCHKR             ; Go Check Item entries                                                           238
      IF T.RETRYSW = 'YES'                ; Need to RETRY Tran                                                              239
        TRANSFER TYPE_1                   ; Yes, Go back to TYPE 1                                                          240
      END IF                                                                                                                  241
     END IF                                                                                                                   242
     END PROC ;TRANERR                                                                                                        243
     ;                                                                                                                        244
-------------------------------------------------------------------------------------------------------------------------------
STMT SEQ  LOG CON .....OPERAND-A.....  OPER  .............OPERAND-B.............  ......RESULT......  PARTIAL FIELD  SEQUENCE   LIST
TYPE NO.  LEV CTR QLF FIELD   SEG-LVL  AIN   QLF FIELD   SEG-LVL  (OR CONSTANT)   QLF FIELD  SEG-LVL  STRT NUMB OPR  73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(PR) (100)        (F,XTRAN   )        (EQ)   (D,1                           )                                      (ASLGEN'D)  245
(PR) (101)                            (NS)     (104                         )                                      (ASLGEN'D)  246
```

Figure D-83 Updating Application: Type 4 Procedure TRANERR (Page 1 of 2)

```
(PR) (102)                          (GO)     (SUB REJRPTS                 )                                (ASLGEN'D)   247
(PR) (103)                          (GO)     (106                         )                                (ASLGEN'D)   248
(PR) (104)                          (R )   (C,AUTO REJECT, SEE DIAG MSG   ) (T,ERRNOTE )                  (ASLGEN'D)   249
(PR) (105)                          (GO)     (SUB REJRPTS                 )                                (ASLGEN'D)   250
(PR) (106)          (X,TRANID  ) 001-1  (EQ)   (C,5ADDITEM                 )                                (ASLGEN'D)   251
(PR) (107)                          (NS)     (END                         )                                (ASLGEN'D)   252
(PR) (108)                          (GO)     (SUB ITEMCHKR                )                                (ASLGEN'D)   253
(PR) (109)          (T,RETRYSW )    (EQ)   (C,YES                         )                                (ASLGEN'D)   254
(PR) (110)                          (NS)     (END                         )                                (ASLGEN'D)   255
(PR) (111)                          (GO)     (TYPE 1                      )                                (ASLGEN'D)   256
```

Figure D-83 Updating Application: Type 4 Procedure TRANERR (Page 2 of 2)

## Sub-Procedure REJRPTS

```
FEB 20, 2002  16.03.15                                                                                    PAGE     13

                                        ****************************
                                        *  PROC NAME - REJRPTS     *
                                        *  INPUT STREAM PROCEDURE  *
                                        ****************************

-------------------------------------------------------------------------------------------------------------------------
STMT REPORT                              MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID        ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM       73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------
(ER)                                                                          (S)                              (ASLGEN'D)   257
    REJRPTS: PROC TYPE SUB                                                                                                 258
    ;                                                                                                                      259
    ; The REJECTED TRANSACTION REPORTs                                                                                     260
    ;  One report for order transaction records                                                                            261
    ;  One report for item  transaction records                                                                            262
    ;                                                                                                                      263
    BLANK: FIELD C 1 HEADING ' '                                                                                           264
    ;                                                                                                                      265
    IF X.TRANID = '3ADDORDR'                                                                                               266
    ;                                                                                                                      267
      REPORT X.TRANID X.CUSTNO X.ORDERNO X.ORDRDATE X.ORDRPERS X.ORDRPONO,                                                 268
             X.ORDRDUED X.ORDRINVG X.ORDRCOMP F.XTRAN F.M4TRAN,                                                            269
             T.ERRNOTE                                                                                                     270
        ITEM X.TRANID,                                                                                                     271
             SPACES 0                                                                                                      272
        ITEM X.CUSTNO X.ORDERNO X.ORDRDATE X.ORDRPERS X.ORDRPONO,                                                          273
             X.ORDRDUED X.ORDRINVG X.ORDRCOMP F.XTRAN F.M4TRAN,                                                            274
             SPACES 1                                                                                                      275
        ITEM F.M4TRAN,                                                                                                     276
             ENDLINE                                                                                                       277
        TITLE 'REJECTED TRANSACTION REPORT - ADD ORDER TRANSACTION RECORDS'                                                278
      END REPORT                                                                                                           279
    END IF                                                                                                                 280
    ;                                                                                                                      281
    IF X.TRANID = '5ADDITEM'                                                                                               282
    ;                                                                                                                      283
      REPORT X.TRANID X.CUSTNO X.ORDERNO X.ITM5ORD1 X.ITM5QTY1,                                                            284
             X.ITM5ORD2 X.ITM5QTY2,                                                                                        285
             X.ITM5ORD3 X.ITM5QTY3,                                                                                        286
             X.ITM5ORD4 X.ITM5QTY4,                                                                                        287
             X.ITM5ORD5 X.ITM5QTY5,                                                                                        288
             F.XTRAN F.M4TRAN T.ERRNOTE                                                                                    289
        ITEM X.TRANID,                                                                                                     290
             SPACES 0                                                                                                      291
        ITEM X.CUSTNO X.ORDERNO X.ITM5ORD1 X.ITM5QTY1 X.ITM5ORD2 X.ITM5QTY2,                                               292
             X.ITM5ORD3 X.ITM5QTY3 X.ITM5ORD4 X.ITM5QTY4,                                                                  293
             X.ITM5ORD5 X.ITM5QTY5 F.XTRAN F.M4TRAN,                                                                       294
             SPACES 1                                                                                                      295
        ITEM F.M4TRAN,                                                                                                     296
             ENDLINE                                                                                                       297
        TITLE 'REJECTED TRANSACTION REPORT - ADD ITEMS TRANSACTION RECORDS'                                                298
      END REPORT                                                                                                           299
    END IF                                                                                                                 300
    ;                                                                                                                      301
    IF X.TRANID NE '3ADDORDR' '5ADDITEM'                                                                                   302
    ;                                                                                                                      303
      REPORT F.M4TRAN X.TRANID X.ALL80                                                                                     304
        TITLE 'REJECTED TRANSACTION REPORT - UNKNOWN TRANSACTION RECORDS'                                                  305
      END REPORT                                                                                                           306
FEB 20, 2002  16.03.15                                                                                    PAGE     14
    END IF                                                                                                                 307
    END PROC ;REJRPTS                                                                                                      308
    ;                                                                                                                      309
-------------------------------------------------------------------------------------------------------------------------
STMT   FIELD    FIELD FLD DEC OUTPT EDIT      INITIAL        LINE 1 OF       LINE 2 OF                          SEQUENCE   LIST
TYPE   NAME     LNGTH TYP PLC EDIT  LGTH       VALUE       COLUMN HEADING  COLUMN HEADING                      73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------
```

Figure D-84 Updating Application: Sub-Procedure REJRPTS (Page 1 of 3)

```
(TF) (BLANK   ) (  1) (C)                                                                                  (ASLGEN'D)   310
------------------------------------------------------------------------------------------------------------------------------
STMT SEQ  LOG CON .....OPERAND-A.....  OPER  ............OPERAND-B............   ......RESULT......  PARTIAL FIELD  SEQUENCE   LIST
TYPE NO.  LEV CTR QLF FIELD   SEG-LVL  ATN   QLF FIELD   SEG-LVL  (OR CONSTANT)   QLF FIELD  SEG-LVL  STRT NUMB OPR  73 TO 80    SEQ
------------------------------------------------------------------------------------------------------------------------------
(PR) (100)        (X,TRANID  ) 001-1  (EQ)   (C,3ADDORDR                       )                                  (ASLGEN'D)   311
(PR) (101)                            (NS)     (103                            )                                  (ASLGEN'D)   312
(PR) (102)                            (GO)     (SUB REPT__7                    )                                  (ASLGEN'D)   313
(PR) (103)        (X,TRANID  ) 001-1  (EQ)   (C,5ADDITEM                       )                                  (ASLGEN'D)   314
(PR) (104)                            (NS)     (106                            )                                  (ASLGEN'D)   315
(PR) (105)                            (GO)     (SUB REPT__8                    )                                  (ASLGEN'D)   316
(PR) (106)        (X,TRANID  ) 001-1  (NE)   (C,3ADDORDR                       )                                  (ASLGEN'D)   317
(PR) (107)                            (NS)     (END                            )                                  (ASLGEN'D)   318
(PR) (108)        (X,TRANID  ) 001-1  (NE)   (C,5ADDITEM                       )                                  (ASLGEN'D)   319
(PR) (109)                            (NS)     (END                            )                                  (ASLGEN'D)   320
(PR) (110)                            (GO)     (SUB REPT__9                    )                                  (ASLGEN'D)   321
FEB 20, 2002  16.03.15                                                                                                  PAGE    15
                                                   *****************************
                                                   *  REQUEST NAME - REPT__7   *
                                                   *  GENERATED REQUEST        *
                                                   *****************************
------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                             MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ        BACK BR RNT AB        SEQUENCE   LIST
TYPE  DATE           REQUESTOR ID      ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM       73 TO 80    SEQ
------------------------------------------------------------------------------------------------------------------------------
(ER)                                                                           (S)                              (ASLGEN'D)   322
------------------------------------------------------------------------------------------------------------------------------
 ST SUM V  8  PG  PG   I SP  MAXIMUM TB COLUMN  DT  PG  STRT LIN SL RH .......SUBFILE........ C E A S R S N      C  SEQUENCE   LIST
 TY RPT S LPI WID HGT  M FRM LPP PGS TO HEADING POS POS PAGE NUM UB PN ENT    NAME   BLK/ FMT H F G T S U D      O  73 TO 80    SEQ
 MP     P              G              LT TYP POS          NUM      ML TD REC           BUF      C C S L T M S      N
------------------------------------------------------------------------------------------------------------------------------
(E1)                                                                                                            (ASLGEN'D)   323
------------------------------------------------------------------------------------------------------------------------------
          SP  Q                 E  N  S  D  C  S  T  C  C  M  M  A  M  D Q PCT/RAT      P                PARTIAL  SEQUENCE   LIST
STMT SEQ  BF  L  FIELD   SEG-   N  P  E  E  T  U  O  U  N  A  I  V  O  E L  FIELD   RA- C     PICTURE    STRT NO  73 TO 80    SEQ
TYPE NO.  COL F  NAME     LVL   D  R  Q  S  L  B  T  M  T  X  N  G  D  C F  NAME     TIO T      EDIT      CHR CHR
------------------------------------------------------------------------------------------------------------------------------
(R1)      ( 0) (X,TRANID  ) 001-1                                                                               (ASLGEN'D)   324
(R1)      ( 1) (X,CUSTNO  ) 001-1                                                                               (ASLGEN'D)   325
(R1)      ( 1) (X,ORDERNO ) 001-1                                                                               (ASLGEN'D)   326
(R1)      ( 1) (X,ORDRDATE) 001-1                                                                               (ASLGEN'D)   327
(R1)      ( 1) (X,ORDRPERS) 001-1                                                                               (ASLGEN'D)   328
(R1)      ( 1) (X,ORDRPONO) 001-1                                                                               (ASLGEN'D)   329
(R1)      ( 1) (X,ORDRDUED) 001-1                                                                               (ASLGEN'D)   330
(R1)      ( 1) (X,ORDRINVG) 001-1                                                                               (ASLGEN'D)   331
(R1)      ( 1) (X,ORDRCOMP) 001-1                                                                               (ASLGEN'D)   332
(R1)      ( 1) (F,XTRAN   )                                                                                     (ASLGEN'D)   333
(R1)      ( 1) (F,M4TRAN  )       (Y)                                                                           (ASLGEN'D)   334
(R1)           (T,ERRNOTE )                                                                                     (ASLGEN'D)   335
------------------------------------------------------------------------------------------------------------------------------
STMT  SEQ                                                                                                       SEQUENCE   LIST
TYPE  NO.                                      TEXT                                                             73 TO 80    SEQ
------------------------------------------------------------------------------------------------------------------------------
(T1)  (101)                 (REJECTED TRANSACTION REPORT - ADD ORDER TRANSACTION RECORDS)                       (ASLGEN'D)   336
(T1)  (102)                 (#                                                          )                       (ASLGEN'D)   337
FEB 20, 2002  16.03.15                                                                                                  PAGE    16
                                                   *****************************
                                                   *  REQUEST NAME - REPT__8   *
                                                   *  GENERATED REQUEST        *
                                                   *****************************
------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                             MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ        BACK BR RNT AB        SEQUENCE   LIST
TYPE  DATE           REQUESTOR ID      ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM       73 TO 80    SEQ
------------------------------------------------------------------------------------------------------------------------------
(ER)                                                                           (S)                              (ASLGEN'D)   338
------------------------------------------------------------------------------------------------------------------------------
 ST SUM V  8  PG  PG   I SP  MAXIMUM TB COLUMN  DT  PG  STRT LIN SL RH .......SUBFILE........ C E A S R S N      C  SEQUENCE   LIST
 TY RPT S LPI WID HGT  M FRM LPP PGS TO HEADING POS POS PAGE NUM UB PN ENT    NAME   BLK/ FMT H F G T S U D      O  73 TO 80    SEQ
 MP     P              G              LT TYP POS          NUM      ML TD REC           BUF      C C S L T M S      N
------------------------------------------------------------------------------------------------------------------------------
(E1)                                                                                                            (ASLGEN'D)   339
------------------------------------------------------------------------------------------------------------------------------
          SP  Q                 E  N  S  D  C  S  T  C  C  M  M  A  M  D Q PCT/RAT      P                PARTIAL  SEQUENCE   LIST
STMT SEQ  BF  L  FIELD   SEG-   N  P  E  E  T  U  O  U  N  A  I  V  O  E L  FIELD   RA- C     PICTURE    STRT NO  73 TO 80    SEQ
TYPE NO.  COL F  NAME     LVL   D  R  Q  S  L  B  T  M  T  X  N  G  D  C F  NAME     TIO T      EDIT      CHR CHR
------------------------------------------------------------------------------------------------------------------------------
(R1)      ( 0) (X,TRANID  ) 001-1                                                                               (ASLGEN'D)   340
(R1)      ( 1) (X,CUSTNO  ) 001-1                                                                               (ASLGEN'D)   341
(R1)      ( 1) (X,ORDERNO ) 001-1                                                                               (ASLGEN'D)   342
(R1)      ( 1) (X,ITM5ORD1) 001-1                                                                               (ASLGEN'D)   343
(R1)      ( 1) (X,ITM5QTY1) 001-1                                                                               (ASLGEN'D)   344
(R1)      ( 1) (X,ITM5ORD2) 001-1                                                                               (ASLGEN'D)   345
(R1)      ( 1) (X,ITM5QTY2) 001-1                                                                               (ASLGEN'D)   346
(R1)      ( 1) (X,ITM5ORD3) 001-1                                                                               (ASLGEN'D)   347
(R1)      ( 1) (X,ITM5QTY3) 001-1                                                                               (ASLGEN'D)   348
(R1)      ( 1) (X,ITM5ORD4) 001-1                                                                               (ASLGEN'D)   349
```

Figure D-84    Updating Application: Sub-Procedure REJRPTS  (Page 2 of 3)

```
(R1)     ( 1)(X,ITM5QTY4) 001-1                                                                              (ASLGEN'D)   350
(R1)     ( 1)(X,ITM5ORD5) 001-1                                                                              (ASLGEN'D)   351
(R1)     ( 1)(X,ITM5QTY5) 001-1                                                                              (ASLGEN'D)   352
(R1)     ( 1)(F,XTRAN   )                                                                                    (ASLGEN'D)   353
(R1)     ( 1)(F,M4TRAN  )      (Y)                                                                           (ASLGEN'D)   354
(R1)         (T,ERRNOTE )                                                                                    (ASLGEN'D)   355
-----------------------------------------------------------------------------------------------------------------------------
STMT  SEQ                                                                                                     SEQUENCE   LIST
TYPE  NO.                                                  TEXT                                               73 TO 80    SEQ
-----------------------------------------------------------------------------------------------------------------------------
(T1)  (101)                 (REJECTED TRANSACTION REPORT - ADD ITEMS TRANSACTION RECORDS)                    (ASLGEN'D)   356
(T1)  (102)                 (#                                                          )                    (ASLGEN'D)   357
FEB 20, 2002  16.03.15                                                                                           PAGE      17
                                                 *****************************
                                                 *  REQUEST NAME - REPT___9  *
                                                 *  GENERATED REQUEST        *
                                                 *****************************
-----------------------------------------------------------------------------------------------------------------------------
STMT REPORT                               MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ         BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID         ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM       73 TO 80    SEQ
-----------------------------------------------------------------------------------------------------------------------------
(ER)                                                                       (S)                               (ASLGEN'D)   358
-----------------------------------------------------------------------------------------------------------------------------
 ST SUM V  8  PG  PG   I SP  MAXIMUM TB COLUMN  DT  PG  STRT LIN SL RH .......SUBFILE........ C E A S R S N      C  SEQUENCE   LIST
 TY RPT S LPI WID HGT  M FRM LPP PGS TO HEADING POS POS PAGE NUM UB PN ENT    NAME   BLK/ FMT H F G T S U D      O  73 TO 80    SEQ
 MP     P              G             LT TYP POS          NUM     ML TD REC           BUF      C C S L T M S      N
-----------------------------------------------------------------------------------------------------------------------------
(E1)                                                                                                         (ASLGEN'D)   359
-----------------------------------------------------------------------------------------------------------------------------
         SP  Q                  E  N  S  D  C  S  T  C  C  M  M  A  M  D Q PCT/RAT      P                PARTIAL  SEQUENCE   LIST
STMT SEQ BF  L  FIELD    SEG-   N  P  E  E  T  U  O  U  N  A  I  V  O  E L  FIELD   RA- C      PICTURE   STRT NO  73 TO 80    SEQ
TYPE NO. COL F  NAME      LVL  D  R  Q  S  L  B  T  M  T  X  N  G  D  C F  NAME    TIO T       EDIT     CHR CHR
-----------------------------------------------------------------------------------------------------------------------------
(R1)         (F,M4TRAN  )                                                                                    (ASLGEN'D)   360
(R1)         (X,TRANID  ) 001-1                                                                              (ASLGEN'D)   361
(R1)         (X,ALL80   ) 001-1                                                                              (ASLGEN'D)   362
-----------------------------------------------------------------------------------------------------------------------------
STMT  SEQ                                                                                                     SEQUENCE   LIST
TYPE  NO.                                                  TEXT                                               73 TO 80    SEQ
-----------------------------------------------------------------------------------------------------------------------------
(T1)  (101)                 (REJECTED TRANSACTION REPORT - UNKNOWN TRANSACTION RECORDS# )                    (ASLGEN'D)   363
```

Figure D-84 Updating Application: Sub-Procedure REJRPTS (Page 3 of 3)

## Sub-Procedure ITEMCHKR

```
FEB 20, 2002  16.03.15                                                                                           PAGE      18
                                                 *****************************
                                                 *  PROC NAME - ITEMCHKR     *
                                                 *  INPUT STREAM PROCEDURE   *
                                                 *****************************
-----------------------------------------------------------------------------------------------------------------------------
STMT REPORT                               MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ         BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID         ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM       73 TO 80    SEQ
-----------------------------------------------------------------------------------------------------------------------------
(ER)                                                                       (S)                               (ASLGEN'D)   364
     ITEMCHKR: PROC SUBROUTINE                                                                                            365
     ;                                                                                                                    366
     ; For the 5ADDITEM Transaction Records                                                                               367
     ;  There are up to 5 ITEMORDER IDs and Quanity sets in the record                                                    368
     ;  We must move the next set to the "first" set fields and RETRY                                                     369
     ;                                                                                                                    370
     LET T.RETRYSW  = 'NO '                                                                                               371
     ;                                                                                                                    372
       IF T.ITMCOUNT = 1                 ; 1st Item/Qty Processed                                                         373
         IF X.ITM5ORD2 NE ' '            ; So chk if 2nd Item/Qty Set                                                     374
           LET X.ITM5ORD1 = X.ITM5ORD2   ; If yes, move data to theyt                                                     375
           LET X.ITM5QTY1 = X.ITM5QTY2   ;  1st set positions                                                             376
           LET T.RETRYSW  = 'YES'        ; Set Retry Switch                                                               377
           RETURN                        ; Exit the Subroutine                                                            378
         ELSE                            ;                                                                                379
           LET T.ITMCOUNT = T.ITMCOUNT + 1 ; If no, increment Item count                                                  380
         END IF                                                                                                           381
       END IF                                                                                                             382
     ;                                                                                                                    383
       IF T.ITMCOUNT = 2                 ; 2nd Item/Qty Processed                                                         384
         IF X.ITM5ORD3 NE ' '            ; So chk if 3rd Item/Qty Set                                                     385
           LET X.ITM5ORD1 = X.ITM5ORD3   ; If yes, move data to the the                                                   386
           LET X.ITM5QTY1 = X.ITM5QTY3   ;  1st set positions                                                             387
           LET T.RETRYSW  = 'YES'        ; Set Retry Switch                                                               388
           RETURN                        ; Exit the Subroutine                                                            389
         ELSE                            ;                                                                                390
```

Figure D-85 Updating Application: Sub-Procedure ITEMCHKR (Page 1 of 2)

```
        LET T.ITMCOUNT = T.ITMCOUNT + 1 ; If no, increment Item count                                                    391
      END IF                                                                                                             392
    END IF                                                                                                               393
  ;                                                                                                                      394
    IF T.ITMCOUNT = 3                 ; 3rd Item/Qty Processed                                                         395
      IF X.ITM5ORD4 NE ' '            ; So chk if 4th Item/Qty Set                                                     396
        LET X.ITM5ORD1 = X.ITM5ORD4   ; If yes, move data to the the                                                   397
        LET X.ITM5QTY1 = X.ITM5QTY4   ;  1st set positions                                                             398
        LET T.RETRYSW  = 'YES'        ; Set Retry Switch                                                               399
        RETURN                        ; Exit the Subroutine                                                            400
      ELSE                            ;                                                                                401
        LET T.ITMCOUNT = T.ITMCOUNT + 1 ; If no, increment Item count                                                  402
      END IF                                                                                                         403
    END IF                                                                                                           404
  ;                                                                                                                  405
    IF T.ITMCOUNT = 4                 ; 4th Item/Qty Processed                                                         406
      IF X.ITM5ORD5 NE ' '            ; So chk if 5th Item/Qty Set                                                     407
        LET X.ITM5ORD1 = X.ITM5ORD5   ; If yes, move data to the the                                                   408
        LET X.ITM5QTY1 = X.ITM5QTY5   ;  1st set positions                                                             409
        LET T.RETRYSW  = 'YES'        ; Set Retry Switch                                                               410
        RETURN                        ; Exit the Subroutine                                                            411
      ELSE                            ;                                                                                412
        LET T.ITMCOUNT = T.ITMCOUNT + 1 ; If no, increment Item count                                                  413
FEB 20, 2002  16.03.15                                                                                    PAGE     19
      END IF                                                                                                         414
    END IF                                                                                                           415
  ;                                                                                                                  416
  IF T.ITMCOUNT = 5           ; ALL 5 ITEM/QTY SETS PROCESSED                                                        417
    LET T.ITMCOUNT = 0        ; YES, WE ARE DONE                                                                     418
    LET T.RETRYSW  = 'NO '    ;                                                                                      419
  END IF                                                                                                             420
  END PROC ;ITEMCHKR                                                                                                 421
  ;                                                                                                                  422
```

| STMT TYPE | SEQ NO. | LOG LEV | CON CTR | .....OPERAND-A..... QLF FIELD | SEG-LVL | OPER ATN | .....OPERAND-B..... QLF FIELD | SEG-LVL | (OR CONSTANT) | .....RESULT..... QLF FIELD | SEG-LVL | PARTIAL FIELD STRT NUMB OPR | SEQUENCE 73 TO 80 | LIST SEQ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| (PR) | (100) | | | | | (R ) | (C,NO | | ) | (T,RETRYSW ) | | | (ASLGEN'D) | 423 |
| (PR) | (101) | | | (T,ITMCOUNT) | | (EQ) | (D,1 | | ) | | | | (ASLGEN'D) | 424 |
| (PR) | (102) | | | | | (NS) | (110 | | ) | | | | (ASLGEN'D) | 425 |
| (PR) | (103) | | | (X,ITM5ORD2) | 001-1 | (NE) | (C) | | | | | | (ASLGEN'D) | 426 |
| (PR) | (104) | | | | | (NS) | (109 | | ) | | | | (ASLGEN'D) | 427 |
| (PR) | (105) | | | | | (R ) | (X,ITM5ORD2) | 001-1 | | (X,ITM5ORD1) | 001-1 | | (ASLGEN'D) | 428 |
| (PR) | (106) | | | | | (R ) | (X,ITM5QTY2) | 001-1 | | (X,ITM5QTY1) | 001-1 | | (ASLGEN'D) | 429 |
| (PR) | (107) | | | | | (R ) | (C,YES | | ) | (T,RETRYSW ) | | | (ASLGEN'D) | 430 |
| (PR) | (108) | | | | | (GO) | (RETURN | | ) | | | | (ASLGEN'D) | 431 |
| (PR) | (109) | | | (T,ITMCOUNT) | | (+ ) | (D,1 | | ) | (T,ITMCOUNT) | | | (ASLGEN'D) | 432 |
| (PR) | (110) | | | (T,ITMCOUNT) | | (EQ) | (D,2 | | ) | | | | (ASLGEN'D) | 433 |
| (PR) | (111) | | | | | (NS) | (119 | | ) | | | | (ASLGEN'D) | 434 |
| (PR) | (112) | | | (X,ITM5ORD3) | 001-1 | (NE) | (C) | | | | | | (ASLGEN'D) | 435 |
| (PR) | (113) | | | | | (NS) | (118 | | ) | | | | (ASLGEN'D) | 436 |
| (PR) | (114) | | | | | (R ) | (X,ITM5ORD3) | 001-1 | | (X,ITM5ORD1) | 001-1 | | (ASLGEN'D) | 437 |
| (PR) | (115) | | | | | (R ) | (X,ITM5QTY3) | 001-1 | | (X,ITM5QTY1) | 001-1 | | (ASLGEN'D) | 438 |
| (PR) | (116) | | | | | (R ) | (C,YES | | ) | (T,RETRYSW ) | | | (ASLGEN'D) | 439 |
| (PR) | (117) | | | | | (GO) | (RETURN | | ) | | | | (ASLGEN'D) | 440 |
| (PR) | (118) | | | (T,ITMCOUNT) | | (+ ) | (D,1 | | ) | (T,ITMCOUNT) | | | (ASLGEN'D) | 441 |
| (PR) | (119) | | | (T,ITMCOUNT) | | (EQ) | (D,3 | | ) | | | | (ASLGEN'D) | 442 |
| (PR) | (120) | | | | | (NS) | (128 | | ) | | | | (ASLGEN'D) | 443 |
| (PR) | (121) | | | (X,ITM5ORD4) | 001-1 | (NE) | (C) | | | | | | (ASLGEN'D) | 444 |
| (PR) | (122) | | | | | (NS) | (127 | | ) | | | | (ASLGEN'D) | 445 |
| (PR) | (123) | | | | | (R ) | (X,ITM5ORD4) | 001-1 | | (X,ITM5ORD1) | 001-1 | | (ASLGEN'D) | 446 |
| (PR) | (124) | | | | | (R ) | (X,ITM5QTY4) | 001-1 | | (X,ITM5QTY1) | 001-1 | | (ASLGEN'D) | 447 |
| (PR) | (125) | | | | | (R ) | (C,YES | | ) | (T,RETRYSW ) | | | (ASLGEN'D) | 448 |
| (PR) | (126) | | | | | (GO) | (RETURN | | ) | | | | (ASLGEN'D) | 449 |
| (PR) | (127) | | | (T,ITMCOUNT) | | (+ ) | (D,1 | | ) | (T,ITMCOUNT) | | | (ASLGEN'D) | 450 |
| (PR) | (128) | | | (T,ITMCOUNT) | | (EQ) | (D,4 | | ) | | | | (ASLGEN'D) | 451 |
| (PR) | (129) | | | | | (NS) | (137 | | ) | | | | (ASLGEN'D) | 452 |
| (PR) | (130) | | | (X,ITM5ORD5) | 001-1 | (NE) | (C) | | | | | | (ASLGEN'D) | 453 |
| (PR) | (131) | | | | | (NS) | (136 | | ) | | | | (ASLGEN'D) | 454 |
| (PR) | (132) | | | | | (R ) | (X,ITM5ORD5) | 001-1 | | (X,ITM5ORD1) | 001-1 | | (ASLGEN'D) | 455 |
| (PR) | (133) | | | | | (R ) | (X,ITM5QTY5) | 001-1 | | (X,ITM5QTY1) | 001-1 | | (ASLGEN'D) | 456 |
| (PR) | (134) | | | | | (R ) | (C,YES | | ) | (T,RETRYSW ) | | | (ASLGEN'D) | 457 |
| (PR) | (135) | | | | | (GO) | (RETURN | | ) | | | | (ASLGEN'D) | 458 |
| (PR) | (136) | | | (T,ITMCOUNT) | | (+ ) | (D,1 | | ) | (T,ITMCOUNT) | | | (ASLGEN'D) | 459 |
| (PR) | (137) | | | (T,ITMCOUNT) | | (EQ) | (D,5 | | ) | | | | (ASLGEN'D) | 460 |
| (PR) | (138) | | | | | (NS) | (END | | ) | | | | (ASLGEN'D) | 461 |
| (PR) | (139) | | | | | (R ) | (D,0 | | ) | (T,ITMCOUNT) | | | (ASLGEN'D) | 462 |
| (PR) | (140) | | | | | (R ) | (C,NO | | ) | (T,RETRYSW ) | | | (ASLGEN'D) | 463 |

Figure D-85 Updating Application: Sub-Procedure ITEMCHKR (Page 2 of 2)

## Type N (Normal) Procedure MAIN

```
FEB 20, 2002  16.03.15                                                                                     PAGE    20
                                                    *****************************
                                                    *  PROC NAME - MAIN         *
                                                    *  INPUT STREAM PROCEDURE   *
                                                    *****************************
-----------------------------------------------------------------------------------------------------------------------
STMT REPORT                                    MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB      SEQUENCE  LIST
TYPE  DATE          REQUESTOR ID              ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM      73 TO 80   SEQ
-----------------------------------------------------------------------------------------------------------------------
(ER)                                                                             (N)                               (ASLGEN'D)   464
     MAIN: PROC TYPE NORMAL                                                                                                     465
     ;                                                                                                                          466
     ; Executed for each Customer Master Record after ALL Updates (if any)                                                      467
     ; have been completed.                                                                                                     468
     ;                                                                                                                          469
     REPORT CUSTNO ORDERNO ORDRDATE ORPERSON ORDPONUM ORDDUDAT,                                                                 470
            ORDINVGN ORDCMPLT ITEMORD ITEMNAME ITMPRICE F.TRAN                                                                  471
       GROUP BY CUSTNO F.TRAN AT LEVEL 1                                                                                        472
       GROUP BY ORDERNO ORDRDATE ORPERSON ORDPONUM,                                                                             473
                ORDDUDAT ORDINVGN ORDCMPLT AT LEVEL 2                                                                           474
       ITEM CUSTNO,                                                                                                             475
            SPACES 0                                                                                                            476
       ITEM ORDPONUM ORDDUDAT ORDINVGN ORDCMPLT,                                                                                477
            ITEMORD ITEMNAME ITMPRICE,                                                                                          478
            SPACES 1                                                                                                            479
       TITLE 'THE NEW CUSTOMER MASTER FILE RECORDS'                                                                             480
     END REPORT                                                                                                                 481
     END PROC                                                                                                                   482
-----------------------------------------------------------------------------------------------------------------------
 ST SUM V  8  PG  PG   I SP  MAXIMUM TB COLUMN  DT  PG  STRT LIN SL RH .......SUBFILE........ C E A S R S N      C  SEQUENCE  LIST
 TY RPT S LPI WID HGT  M FRM LPP PGS TO HEADING POS POS PAGE NUM UB PN ENT    NAME  BLK/ FMT H F G T S U D      O  73 TO 80   SEQ
 MP    P               G              LT TYP POS        NUM      ML TD REC            BUF     C C S L T M S      N
-----------------------------------------------------------------------------------------------------------------------
(E1)                                                                                                               (ASLGEN'D)   483
-----------------------------------------------------------------------------------------------------------------------
          SP  Q              E  N  S  D  C  S  T  C  C  M  M  A  M  D Q PCT/RAT     P                   PARTIAL  SEQUENCE  LIST
STMT SEQ  BF  L  FIELD  SEG-  N  P  E  E  T  U  O  U  N  A  I  V  O  E L  FIELD   RA- C      PICTURE     STRT NO  73 TO 80   SEQ
TYPE NO.  COL F  NAME    LVL  D  R  Q  S  L  B  T  M  T  X  N  G  D  C F  NAME    TIO T       EDIT       CHR CHR
-----------------------------------------------------------------------------------------------------------------------
(R1)     ( 0)  (CUSTNO  ) 001-1          (1)                                                                (ASLGEN'D)   484
(R1)           (ORDERNO ) 002-2          (2)                                                                (ASLGEN'D)   485
(R1)           (ORDRDATE) 002-2          (2)                                                                (ASLGEN'D)   486
(R1)           (ORPERSON) 002-2          (2)                                                                (ASLGEN'D)   487
(R1)     ( 1)  (ORDPONUM) 002-2          (2)                                                                (ASLGEN'D)   488
(R1)     ( 1)  (ORDDUDAT) 002-2          (2)                                                                (ASLGEN'D)   489
(R1)     ( 1)  (ORDINVGN) 002-2          (2)                                                                (ASLGEN'D)   490
(R1)     ( 1)  (ORDCMPLT) 002-2          (2)                                                                (ASLGEN'D)   491
(R1)     ( 1)  (ITEMORD ) 005-3                                                                             (ASLGEN'D)   492
(R1)     ( 1)  (ITEMNAME) 005-3                                                                             (ASLGEN'D)   493
(R1)     ( 1)  (ITMPRICE) 005-3                                                                             (ASLGEN'D)   494
(R1)         (F,TRAN    )                (1)                                                                (ASLGEN'D)   495
-----------------------------------------------------------------------------------------------------------------------
STMT  SEQ                                                                                                    SEQUENCE  LIST
TYPE  NO.                                     TEXT                                                           73 TO 80   SEQ
-----------------------------------------------------------------------------------------------------------------------
(T1)  (101)               (THE NEW CUSTOMER MASTER FILE RECORDS#                      )                    (ASLGEN'D)   496
```

Figure D-86    Updating Application: Type N (Normal) Procedure MAIN

## Transaction Diagnostic Messages

```
FEB 20, 2002  16.03.15                                                                                     PAGE    21
** MK4TG05  TYPE 1  TRANSACTION IDENTIFIER 3ADDORDR - FIELD VIOLATES INPUT EDIT PATTERN SPECIFICATIONS. FIELD LOC    9       *****
                    M4TRAN   RECORD       7 - KEY 1 - 0001A
** MK4TG05  TYPE 1  TRANSACTION IDENTIFIER 3ADDORDR - FIELD VIOLATES INPUT EDIT PATTERN SPECIFICATIONS. FIELD LOC   25       *****
                    M4TRAN   RECORD      10 - KEY 1 - 00115
** MK4RL01  TYPE 1  TRANSACTION IDENTIFIER 3ADDORDR - RECORD NOT ON MASTER FILE AND CREATE NOT SPECIFIED.                    *****
                    M4TRAN   RECORD      13 - KEY 1 - 00638
                    M4OLD    RECORD       8 - KEY 1 - 00731
                    M4NEW    RECORD       7 - KEY 1 - 00551
** MK4RL01  TYPE 1  TRANSACTION IDENTIFIER 5ADDITEM - RECORD NOT ON MASTER FILE AND CREATE NOT SPECIFIED.                    *****
                    M4TRAN   RECORD      20 - KEY 1 - 00777
                    M4OLD    RECORD      10 - KEY 1 - 00818
                    M4NEW    RECORD       9 - KEY 1 - 00738
** MK4TG06  TYPE 1  TRANSACTION IDENTIFIER 3ADDORDR - DATE FIELD IS INVALID. FIELD LOC IS    19                          *****
                    M4TRAN   RECORD      26 - KEY 1 - 00818
** MK4TG06  TYPE 1  TRANSACTION IDENTIFIER 3ADDORDR - DATE FIELD IS INVALID. FIELD LOC IS    33                          *****
                    M4TRAN   RECORD      27 - KEY 1 - 00818
```

Figure D-87    Updating Application: Transaction Diagnostic Messages

## Updating Run Statistics

```
FEB 20, 2002  16.03.15                                                                                    PAGE     22
** MK4W204  TYPE 0  NUMBER OF MESSAGES PRINTED IS    6.                                                          *****
                    M4OLD    -       15 RECORDS INPUT
                    M4NEW    -       15 RECORDS OUTPUT
                    M4TRAN   -       35 RECORDS INPUT
                    M4CORD1  -       26 RECORDS INPUT
                    M4REPO   -      229 RECORDS OUTPUT
                    M4INPUT  -      277 RECORDS INPUT
                    M4LIST   -      869 RECORDS OUTPUT
                      15 TRACKS ASSIGNED TO M4LIB    --   13 TRACKS NOT FULL. LIBRARY DIRECTORY BLOCKING FACTOR IS  800.
** MK4W209  TYPE 0  1002840 BYTES OF MAIN STORAGE UNUSED DURING DECODING PHASE                                  *****
                     997038 BYTES OF MAIN STORAGE UNUSED DURING COMPILATION PHASE
                     564944 BYTES OF MAIN STORAGE UNUSED DURING PROCESSING PHASE
```

Figure D-88 Updating Application: Updating Run Statistics

## Listing of Transaction Records Report

```
** MK4ED02  TYPE 0  REQUEST-REPT___2  REPORT-1  REQUESTOR ID-                                                  *****
                                    LISTING OF TRANSACTION RECORDS 80/80 WITH RECORD COUNT                  PAGE 1
 ------------------------------------------------------------------------------------------------------------------------------
       M4TRAN                                        TRAN RECORD
                                                   ALL 80 POSITIONS
 ------------------------------------------------------------------------------------------------------------------------------
                1  3ADDORDR0000199000062001JOE12345030302NN
                2  5ADDITEM0000199000G46     00001M33     00003M56     00002D77     00001F74     00001
                3  3ADDORDR0000299001062001BOB6789A120401NN
                4  5ADDITEM0000299001H98     00003ERR     00099
                5  3ADDORDR0000299002062101APCB531A020502NN
                6  5ADDITEM0000299002G46     00002
                7  3ADDORDR0001A99000062001ERR12345030302NN
                8  3ADDORDR0011599014082701RAN12543043002NN
                9  5ADDITEM0011599014L99     00003013     00002M56     00000
               10  3ADDORDR001159991408270199912543043002NN
               11  3ADDORDR0034399018062001KENB1632070601NN
               12  5ADDITEM0034399018K69     00001
               13  3ADDORDR0063899006062794NOM99953020102NN
               14  3ADDORDR0073199012052901RANB3219030302NN
               15  4ADDITEM0073199012019     00003872     00002
               16  5ADDITEM0073199012019     00003872     00002
               17  3ADDORDR0073199019071701RANC1399101301NN
               18  5ADDITEM0073199019027     00001072     00004ERR     00000
               19  3ADDORDR0073899006062794TOD99953020102NN
               20  5ADDITEM0077799006ERR     00009
               21  3ADDORDR0081899007052101JJKABC81091501NN
               22  5ADDITEM0081899007H61     00008041     00007
               23  3ADDORDR0081899020071501MBPBING1101901NN
               24  3ADDERRR0081899020071501MBPBING1101901NN
               25  5ADDITEM0081899020D77     00006E56     00005E66     00004
               26  3ADDORDR0081899920171501MBPBING1101901NN
               27  3ADDORDR0081899921071501MBPBING1131901NN
               28  3ADDORDR0084799008062101KREX0C81081501NN
               29  5ADDITEM0084799008F74     00003
               30  3ADDORDR0084799009062201JONX0C82081901NN
               31  5ADDITEM0084799009F91     00001M57     00002
               32  3ADDORDR0091899010071301BOBBC123111901NN
               33  5ADDITEM0091899010F87     00001L99     00003M50     00002
               34  3ADDORDR0100099022080501IKE99B51033102NN
               35  5ADDITEM0100099022072     00002
** MK4FT03  TYPE 0  END OF REPORT.                                                                             *****
```

Figure D-89 Updating Application: Listing of Transaction Records Report

## Listing of Orders that Pre-Date Master File Order Dates

```
** MK4ED02  TYPE 0  REQUEST-REPT__5  REPORT-1  REQUESTOR ID-                                                                  *****
                                 LIST OF ORDERS WITH ORDER DATES EARILER THAN MASTER FILE ORDER DATES                  PAGE 1
 ------------------------------------------------------------------------------------------------------------------------------
   TRANS    CUST   ORDER     TRAN       MASTER        M4TRAN
    ID      NO.    NO.      DATE        DATE
 ------------------------------------------------------------------------------------------------------------------------------
  3ADDORDR  00001  99000  06/20/2001  08/01/2001              1
  3ADDORDR  00343  99018  06/20/2001  07/11/2001             11
  3ADDORDR  00343  99018  06/20/2001  08/01/2001             11
  3ADDORDR  00738  99006  06/27/1994  01/14/2001             19
** MK4FT03  TYPE 0  END OF REPORT.                                                                                              *****
```

Figure D-90 Updating Application: Listing of Orders that Pre-Date Master File Order Dates

## Listing of Update Customer Records

```
** MK4ED02  TYPE 0  REQUEST-REPT___6  REPORT-1  REQUESTOR ID-                                                                  *****
                                                THE UPDATED CUSTOMER MASTER FILE RECORDS                                  PAGE 1
 ------------------------------------------------------------------------------------------------------------------------------
 CUSTOMER  ORDER   ORDER   SALES  PURCH. ORDER  DUE    INVOICE     ORDER    STOCK NO.     DESCRIPTION        PRICE      STRAN
  NUMBER   NUMBER  DATE    PERSON    NUMBER     DATE  GENERATED? COMPLETED?  ORDERED
 ------------------------------------------------------------------------------------------------------------------------------
  00001    99000   062001   JOE      12345     030302    N          N            -                  -        $-  MI
                                                                                G46      V:TRAN UTILITY GDE      15.00  MMI
                                                                                M33      V:TRAN 7.5 CHG PAGES     5.00  MMI
                                                                                M56      2" BINDER NOTEBOOK       7.50  MMI
                                                                                D77      V:TRAN CLASS GDE-A      97.50  MMI
                                                                                F74      V:TRAN CLASS GDE-C     103.50  MMI
  00002    99001   062001   BOB      6789A     120401    N          N            -                  -        $-  MI
                                                                                H98      OB6789A120481NN8        15.00  MMI
           99002   062101   APC      B531A     020502    N          N            -                  -        $-  MI
                                                                                G46      V:TRAN UTILITY GDE      15.00  MMI
  00115    99014   082701   RAN      12543     043002    N          N            -                  -        $-  MI
                                                                                L99      V:BLDR ENVIRON GDE-O    17.50  MMI
                                                                                013      V:BLDR REF SUMMARY      51.00  MMI
  00343    99018   062001   KEN      B1632     070601    N          N            -                  -        $-  MI
                                                                                K69      EEO REPORT GUIDE        67.00  MMI
  00731    99012   052901   RAN      B3219     030302    N          N            -                  -        $-  MI
                                                                                019      V:BLDR ENV GUIDE-VSE    51.00  MMI
                                                                                872      G/L REFERENCE GUIDE     75.00  MMI
           99019   071701   RAN      C1399     101301    N          N            -                  -        $-  MI
                                                                                027      V:BLDR CLASS GUIDE-A    55.00  MMI
                                                                                072      V:BLDR CLASS GUIDE-B    55.00  MMI
  00738    99006   062794   TOD      99953     020102    N          N            -                  -        $-  MI
  00818    99007   052101   JJK      ABC81     091501    N          N            -                  -        $-  MI
                                                                                H61      V:BLDR INST GDE-VSE     15.00  MMI
                                                                                041      V:BLDR PROCEDURE GDE    55.00  MMI
           99020   071501   MBP      BING1     101901    N          N            -                  -        $-  MI
                                                                                D77      V:TRAN CLASS GDE-A      97.50  MMI
                                                                                E56      V:BLDR USER I/O GDE     27.50  MMI
                                                                                E66      V:BLDR GDBI GUIDE       37.50  MMI
  00847    99008   062101   KRE      X0C81     081501    N          N            -                  -        $-  MI
                                                                                F74      V:TRAN CLASS GDE-C     103.50  MMI
           99009   062201   JON      X0C82     081901    N          N            -                  -        $-  MI
                                                                                F91      V:TRAN ADM GUIDE        30.00  MMI
                                                                                M57      3" BINDER NOTEBOOK       9.50  MMI
  00918    99010   071301   BOB      BC123     111901    N          N            -                  -        $-  MI
                                                                                F87      V:TRAN CLASS GDE-B      97.50  MMI
                                                                                L99      V:BLDR ENVIRON GDE-O    17.50  MMI
                                                                                M50      V:TRAN SECURITY GDE     27.50  MMI
  01000    99022   080501   IKE      99B51     033102    N          N            -                  -        $-  MI
                                                                                072      V:BLDR CLASS GUIDE-B    55.00  MMI
** MK4FT06  TYPE 1        42 FIELDS WERE PRINTED AS INVALID, MISSING OR OVERFLOW.                                              *****
** MK4FT03  TYPE 0  END OF REPORT.                                                                                              *****
```

Figure D-91 Updating Application: Listing of Update Customer Records

## Listing of Rejected Add Order Transactions

```
** MK4ED02  TYPE 0  REQUEST-REPT___7  REPORT-1  REQUESTOR ID-                                                                  *****
                                  REJECTED TRANSACTION REPORT - ADD ORDER TRANSACTION RECORDS                          PAGE 1
-------------------------------------------------------------------------------------------------------------------------------
 TRANS   CUST  ORDER ORDER  ORDER  ORDER  DUE   INV INV XTRAN     M4TRAN
  ID     NO.    NO.  DATE   PERSON  PO    DATE  GEN CMP
-------------------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------------------------------
           ERROR
          COMMENT
-------------------------------------------------------------------------------------------------------------------------------
3ADDORDR 0001A 99000 062001  ERR   12345 030302  N   N     7              7
  AUTO REJECT, SEE DIAG MSG
3ADDORDR 00115 99914 082701  999   12543 043002  N   N     7             10
  AUTO REJECT, SEE DIAG MSG
3ADDORDR 00638 99006 062794  NOM   99953 020102  N   N     8             13
  AUTO REJECT, SEE DIAG MSG
3ADDORDR 00818 99920 171501  MBP   BING1 101901  N   N     6             26
  AUTO REJECT, SEE DIAG MSG
3ADDORDR 00818 99921 071501  MBP   BING1 131901  N   N     6             27
  AUTO REJECT, SEE DIAG MSG
** MK4FT03  TYPE 0  END OF REPORT.                                                                                            *****
```

Figure D-92 Updating Application: Listing of Rejected Add Order Transactions

## Listing of Rejected Add Item Transactions

```
** MK4ED02  TYPE 0  REQUEST-REPT___8  REPORT-1  REQUESTOR ID-                                                                  *****
                                  REJECTED TRANSACTION REPORT - ADD ITEMS TRANSACTION RECORDS                          PAGE 1
-------------------------------------------------------------------------------------------------------------------------------
 TRANS   CUST  ORDER  ITEM    ITEM    ITEM    ITEM    ITEM    ITEM    ITEM    ITEM    ITEM    ITEM   XTRAN     M4TRAN
  ID     NO.    NO.  ORDR 1   QTY 1  ORDR 2   QTY 2  ORDR 3   QTY 3  ORDR 4   QTY 4  ORDR 5   QTY 5
-------------------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------------------------------
           ERROR
          COMMENT
-------------------------------------------------------------------------------------------------------------------------------
5ADDITEM 00002 99001 ERR          99 ERR          99                                                                  4
  ITEM ORDERED NOT ON FILE
5ADDITEM 00115 99014 M56            013           2 M56                                                   1              9
  QTY INVALID FROM SET 3
5ADDITEM 00731 99019 ERR            072           4 ERR                                                   1             18
  QTY INVALID FROM SET 3
5ADDITEM 00777 99006 ERR           9                                                                     8             20
  AUTO REJECT, SEE DIAG MSG
** MK4FT03  TYPE 0  END OF REPORT.                                                                                            *****
```

Figure D-93 Updating Application: Listing of Rejected Add Item Transactions

## Listing of Rejected Unknown Transactions

```
** MK4ED02  TYPE 0  REQUEST-REPT___9  REPORT-1  REQUESTOR ID-                                                                  *****
                                  REJECTED TRANSACTION REPORT - UNKNOWN TRANSACTION RECORDS                            PAGE 1
-------------------------------------------------------------------------------------------------------------------------------
      M4TRAN       TRANS                                    TRAN RECORD
                    ID                                    ALL 80 POSITIONS
-------------------------------------------------------------------------------------------------------------------------------
              15  4ADDITEM  4ADDITEM0073199012019    00003872    00002
              24  3ADDERRR  3ADDERRR0081899020071501MBPBING1101901NN
** MK4FT03  TYPE 0  END OF REPORT.                                                                                            *****
```

Figure D-94 Updating Application: Listing of Rejected Unknown Transactions

## Listing of New Customer File Records

```
** MK4ED02  TYPE 0  REQUEST-MAIN      REPORT-1  REQUESTOR ID-                                                    *****
                                  THE NEW CUSTOMER MASTER FILE RECORDS                                    PAGE 1

CUSTOMER  ORDER   ORDER   SALES  PURCH. ORDER  DUE    INVOICE    ORDER     STOCK NO.     DESCRIPTION          PRICE    TRAN
 NUMBER   NUMBER  DATE    PERSON    NUMBER     DATE  GENERATED? COMPLETED?  ORDERED

 00001    10001   010401  PEG       A0001    100101    N          N        G38      1" BINDER NOTEBOOK      $5.00    1
                                                                           G40      V:BLDR REFERENCE GDE    35.00
          10002   080101  BRR       A0002    110101    N          N        M14      V:BLDR ASL REF GDE      $31.00
                                                                           M23      V:TRAN SCREEN GDE       25.00
          99000   062001  JOE       12345    030302    N          N        D77      V:TRAN CLASS GDE-A      $97.50
                                                                           F74      V:TRAN CLASS GDE-C      103.50
                                                                           G46      V:TRAN UTILITY GDE      15.00
                                                                           M33      V:TRAN 7.5 CHG PAGES    5.00
                                                                           M56      2" BINDER NOTEBOOK      7.50
 00002    99001   062001  BOB       6789A    120401    N          N        H98      OB6789A120481NN8        $15.00   1
          99002   062101  APC       B531A    020502    N          N        G46      V:TRAN UTILITY GDE      $15.00
 00115    11501   071201  PEG       A1151    073001    Y          Y        010      ACCOUNTS PAYABLE UM     $45.00   5
          11502   072201  PEG       A1152    083101    Y          N        072      V:BLDR CLASS GUIDE-B    $55.00
                                                                           106      V:BLDR TABLE TIPS       5.00
                                                                           110      V:BLDR REPORT TIPS      5.00
          99014   082701  RAN       12543    043002    N          N        L99      V:BLDR ENVIRON GDE-O    $17.50
                                                                           013      V:BLDR REF SUMMARY      51.00
 00131    31135   050501  JAM       K123J    070101    Y          Y        D77      V:TRAN CLASS GDE-A      $97.50
          31735   070701  JAM       K132J    080101    Y          Y        027      V:BLDR CLASS GUIDE-A    $55.00
 00294    29401   042901  JEE       B2941    060101    Y          Y        978      PURCHASE ORDER GDE      $35.00
 00343    34301   071101  ANS       A3431    080301    Y          Y        865      V:INFM REFERENCE GDE    $39.00   1
                                                                           866      V:INFM INSTALL GDE      19.00
                                                                           873      V:INFM OPS GUIDE        19.00
          34302   080101  PEG       A3423    090501    N          N        852      PURCHASE ORDER ENVR     $24.00
                                                                           872      G/L REFERENCE GUIDE     75.00
          99018   062001  KEN       B1632    070601    N          N        K69      EEO REPORT GUIDE        $67.00
 00551    51135   060601  JAM       K123J    070101    Y          Y        D77      V:TRAN CLASS GDE-A      $97.50
          51755   070901  JAM       K132J    080901    Y          Y        027      V:BLDR CLASS GUIDE-A    $55.00
 00731    99012   052901  RAN       B3219    030302    N          N        019      V:BLDR ENV GUIDE-VSE    $51.00   1
                                                                           872      G/L REFERENCE GUIDE     75.00
          99019   071701  RAN       C1399    101301    N          N        027      V:BLDR CLASS GUIDE-A    $55.00
                                                                           072      V:BLDR CLASS GUIDE-B    55.00
 00738    73891   011401  JEF       B7381    012901    Y          Y        L06      V:INFM CLASS PACK       $299.99  1
                                                                           L06F     V:INFM CLASS SLIDES     99.00
                                                                           L96      V:BLDR APPL GUIDE       50.00
          99006   062794  TOD       99953    020102    N          N           -                -            $-
 00818    99007   052101  JJK       ABC81    091501    N          N        H61      V:BLDR INST GDE-VSE     $15.00   5
                                                                           041      V:BLDR PROCEDURE GDE    55.00
          99020   071501  MBP       BING1    101901    N          N        D77      V:TRAN CLASS GDE-A      $97.50
                                                                           E56      V:BLDR USER I/O GDE     27.50
                                                                           E66      V:BLDR GDBI GUIDE       37.50
 00847    99008   062101  KRE       X0C81    081501    N          N        F74      V:TRAN CLASS GDE-C      $103.50  1
          99009   062201  JON       X0C82    081901    N          N        F91      V:TRAN ADM GUIDE        $30.00
                                                                           M57      3" BINDER NOTEBOOK      9.50
                                  THE NEW CUSTOMER MASTER FILE RECORDS                                    PAGE 2

CUSTOMER  ORDER   ORDER   SALES  PURCH. ORDER  DUE    INVOICE    ORDER     STOCK NO.     DESCRIPTION          PRICE    TRAN
 NUMBER   NUMBER  DATE    PERSON    NUMBER     DATE  GENERATED? COMPLETED?  ORDERED

 00867    86731   061501  WAV       W1531    100101    Y          Y        013      V:BLDR REF SUMMARY      $51.00
          86751   071501  SUR       S1552    110901    Y          Y        041      V:BLDR PROCEDURE GDE    $55.00
 00918    91801   051201  GEE       A9181    071501    Y          Y        L97      V:INFM QUERY GUIDE      $45.00   1
                                                                           M01      V:TRAN ENV GDE CICS     47.50
          99010   071301  BOB       BC123    111901    N          N        F87      V:TRAN CLASS GDE-B      $97.50
                                                                           L99      V:BLDR ENVIRON GDE-O    17.50
                                                                           M50      V:TRAN SECURITY GDE     27.50
 00951    91395   090601  JOH       9123A    010902    N          N        013      V:BLDR REF SUMMARY      $51.00
          91795   090901  UNC       9132B    010902    N          N        041      V:BLDR PROCEDURE GDE    $55.00
 01000    99022   080501  IKE       99B51    033102    N          N        072      V:BLDR CLASS GUIDE-B    $55.00   1
** MK4FT06  TYPE 1         3 FIELDS WERE PRINTED AS INVALID, MISSING OR OVERFLOW.                                *****
** MK4FT03  TYPE 0  END OF REPORT.                                                                              *****
```

Figure D-95 Updating Application: Listing of New Customer File Records

# Header Trailer Application

## Source Code and Report Output

```
FEB 20, 2002  16.03.15                                                                                                PAGE      1

                                                ******************************************
                                                *  VISION:Builder   4400  (OS/390 - 14.0)  *
                                                *           COPYRIGHT (C) 2002             *
                                                *  COMPUTER ASSOCIATES INTERNATIONAL, INC. *
                                                ******************************************
VISION:Builder JCR1 14.0 022202                    ENABLED FOR IBM LANGUAGE ENVIRONMENT                    BUILD STAMP = 102053,17:28:40.
                                                COMLIB  MODIFICATIONS (PTFs, APARs) INSTALLED
                                             201,202,203,204,205,206,207,208,209,210,212,213,214,215,216

=============================================================================================================================
=                                                                                                                           =
=                    I N S T A L L A T I O N    P A R A M E T E R S    ( M 4 P A R A M S , M A R K L I B P )               =
=                                                                                                                           =
=  SYSTEM DELIMITER:   #          PAGE HEIGHT:       66        M4LIST WIDTH:      132       DEF WIDTH OF PAGE:  0         =
=                                                                                                                           =
=  AUTO GRAND:         N          HEADING CHAR:      -         SUBTITLE REPEAT:   N         INVALID FIELD:      *         =
=                                                                                                                           =
=  MISSING FIELD:      -          NON-EDIT FIELD:    +         PERCENT CHAR:      %         LEFT SEPARATOR:     (         =
=                                                                                                                           =
=  RIGHT SEPARATOR:    )          SINGLE SEPARATOR:  ,         SOURCE SPACING:    1         PRINT MESSAGES:     Y         =
=                                                                                                                           =
=  CONSOLE MESSAGES:   N          M4REPO BLOCKSIZE:  4,096     INPUT I/O BUFFERS: 2         OUTPUT I/O BUFFERS: 1         =
=                                                                                                                           =
=  SNGL-STEP STORAGE:  8,192      SNGL-STEP SORTSIZE: 524,288  DIGIT SELECT CHAR: 9         ZERO SUPPRESS CHAR: Z         =
=                                                                                                                           =
=  CURRENCY CHAR:      $          PLUS CHAR:         +         MINUS CHAR:        -         CHECK PROTECT CHAR: *         =
=                                                                                                                           =
=  DECIMAL CHAR:       .          GROUPING CHAR:     ,         PRIMARY PLOT CHAR: X         SECONDARY PLOT CHAR:*         =
=                                                                                                                           =
=  FIT PLOT CHAR:      .          HORIZONTAL AXIS:   _         HORIZONTAL HASH:   |         VERTICAL AXIS:     |         =
=                                                                                                                           =
=  VERTICAL HASH:      -          MINUTES/HOUR:      60        SECONDS/MINUTE:    60        TIME DELIMITER:     HH:MM:SS  =
=                                                                                                                           =
=  DATE FORMAT:      MM DD, YYYY  TODAY FORMAT+DELIM: MM/DD/YY  ISDATE DELIMITER:  YYYY-MM-DD JULIAN DELIMITER:  YY.DDD    =
=                                                                                                                           =
=  SORT PROGRAM CODE:  2          MINCORE VALUE:     12 K      ALT M4LIST WIDTH:  264       ALT DEF W/OF PAGE:  0         =
=                                                                                                                           =
=  MAX LINES OF TRACE: 1,024      ITEM TRACKING:     0         SUPPRESS NDS REPT?: N        DEFAULT MAXGETMN:   1,024 K   =
=                                                                                                                           =
=  CONDITION CODE 1:   0          CONDITION CODE 2:  4         CONDITION CODE 3:  8         CONDITION CODE 4:   16        =
=                                                                                                                           =
=  ISAM INDEX INCORE:  N          M4LIB BLKG FACTOR: 0         M4LIB RESERVE:     0         M4LIB COMPONENT:    C4.5 B/V  =
=                                                                                                                           =
=  DEFAULT F/P AMODE:  24         M4LIB COMPRESSION?: N        M4LIB COMPRESS MIN: 507                                    =
=                                                                                                                           =
=============================================================================================================================

FEB 20, 2002  16.03.15                                                                                                PAGE      2
-----------------------------------------------------------------------------------------------------------------------------
   RUN     STMT    FILE    O N S D U T R A B S D S L  RB  R S M R S          START          END            SEQUENCE    LIST
   NAME    TYPE    NAME    L E E I P R E U U R L C S  PL  J O O F S          SEARCH         SEARCH         73 TO 80    SEQ
                           D W Q R D N P D F T M N T  TK  T P P O R
-----------------------------------------------------------------------------------------------------------------------------
(UPDCUSIM) (RC) (HDITRMSTR)  (S)     (S,S)      (T,G)         (R)                                        (ASLGEN'D)     1
    CONTROL NAME UPDCUSIM TERM LISTGEN                                                                                    2
    FILE MASTER OUTPUT HDITRMSTR                                                                                          3
    FILE TRAN   HDITRATRN GROUPS HDITRGRP                                                                                 4
    FILE REPORT                                                                                                           5
    ;                                                                                                                     6
-----------------------------------------------------------------------------------------------------------------------------
   RUN     STMT  LOGICAL  LBL  DTF/DD     KEY       KEY       KEY     QUALIFIER   USER I/O  ARRAY      PASSWORD/   SEQUENCE    LIST
   NAME    TYPE  FILENAME       NAME     FIELD-1   FIELD-2   FIELD-3  ICF CRD DBI   MODULE   QUAL       AUTH.-ID    73 TO 80    SEQ
-----------------------------------------------------------------------------------------------------------------------------
(UPDCUSIM) (RF) (HDITRMSTR) (S) (M4NEW   )                                                                (ASLGEN'D)     7
(UPDCUSIM) (RF) (HDITRATRN) (S) (M4TRAN  )                                                                (ASLGEN'D)     8
-----------------------------------------------------------------------------------------------------------------------------
   RUN     STMT                TRANSACTION GROUPS                                                        SEQUENCE    LIST
   NAME    TYPE     1         2         3         4         5                                            73 TO 80    SEQ
-----------------------------------------------------------------------------------------------------------------------------
(UPDCUSIM) (RG) (HDITRGRP )                                                                               (ASLGEN'D)     9



=============================================================================================================================
=                                                                                                                           =
=                                                F I L E    S U M M A R Y                                               =
=                                                                                                                           =
```

Figure D-96 Updating Application: Source Code and Report Output (Page 1 of 4)

```
= JOBNAME: KREJO04X                                  STEPNAME: PROC2                                                  =
=                                                                                                                     =
=  FILE NAME: M4NEW                                  DSNAME: BUILDER.R140.HDRTRLR.MSTR.R15#1                          =
=  FILE NAME: M4TRAN                                 DSNAME: KREJO04.KREJO04X.JOB07498.D0000103.?                     =
=  FILE NAME: M4LIB                                  DSNAME: BUILDER.R140.M4LIB.R15#1                                 =
=                                                                                                                     =
=======================================================================================================================
FEB 20, 2002  16.03.15                                                                                         PAGE     3
                                                       ****************************
                                                       *  PROC NAME - TRANPREP    *
                                                       *  INPUT STREAM PROCEDURE  *
                                                       ****************************
-----------------------------------------------------------------------------------------------------------------------
STMT REPORT                                 MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ         BACK BR RNT AB       SEQUENCE  LIST
TYPE  DATE          REQUESTOR ID           ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM       73 TO 80   SEQ
-----------------------------------------------------------------------------------------------------------------------
(ER)                                                                           (1)                                 (ASLGEN'D)    10
     TRANPREP: PROC TYPE1                                                                                                        11
     ;                                                                                                                           12
     ; This TYPE 1 Procedure is executed for each new transaction record                                                         13
     ;                                                                                                                           14
     ;   For Headers, save custno for trailers A & B                                                                             15
     ;       Headers, save custno for trailers A & B                                                                             16
     ;   For Trailer A & B, put saved custno into record                                                                         17
     ;                                                                                                                           18
     ; List the input records                                                                                                    19
     ;                                                                                                                           20
     SAVECUST: FIELD C 5  HEADING 'SAVED' 'CUSTNO'                                                                               21
     ;                                                                                                                           22
     DO CASE                                                                                                                     23
      CASE WHERE X.TRAILID = 'A'         ; Trailer rcd A                                                                         24
       LET X.COMMCUST = T.SAVECUST       ; Yes, CUSTNO to common area                                                            25
      CASE WHERE X.TRAILID = 'B'         ; Trailer rcd B                                                                         26
       LET X.COMMCUST = T.SAVECUST       ; Yes, CUSTNO to common area                                                            27
      ELSE                                                                                                                       28
       LET T.SAVECUST = X.CUSTNO         ; Save CUSTNO from header                                                               29
       LET X.COMMCUST = T.SAVECUST       ; put CUSTNO into common area                                                           30
     END DO                                                                                                                      31
     ;                                                                                                                           32
     REPORT F.M4TRAN X.ALL80           ; No, list Tran Rcd                                                                       33
     TITLE 'LISTING OF HEADER AND TRAILER RECORDS'                                                                               34
     END REPORT                                                                                                                  35
     ;                                                                                                                           36
     END PROC ;TRANPREP                                                                                                          37
     ;                                                                                                                           38
-----------------------------------------------------------------------------------------------------------------------
STMT  FIELD    FIELD FLD DEC OUTPT EDIT     INITIAL       LINE 1 OF      LINE 2 OF                                  SEQUENCE  LIST
TYPE  NAME     LNGTH TYP PLC EDIT  LGTH      VALUE      COLUMN HEADING  COLUMN HEADING                              73 TO 80   SEQ
-----------------------------------------------------------------------------------------------------------------------
(TF) (SAVECUST) (  5) (C)                              (SAVED         ) (CUSTNO        )                           (ASLGEN'D)    39
-----------------------------------------------------------------------------------------------------------------------
STMT SEQ  LOG CON .....OPERAND-A.....  OPER  ............OPERAND-B............   ......RESULT......  PARTIAL FIELD  SEQUENCE  LIST
TYPE NO.  LEV CTR QLF FIELD  SEG-LVL  ATN   QLF FIELD  SEG-LVL  (OR CONSTANT)   QLF FIELD  SEG-LVL  STRT NUMB OPR  73 TO 80   SEQ
-----------------------------------------------------------------------------------------------------------------------
(PR) (100)        (X,TRAILID ) 001-1  (EQ)  (C,A                               )                                   (ASLGEN'D)    40
(PR) (101)                            (NS)    (104                             )                                   (ASLGEN'D)    41
(PR) (102)                            (R )  (T,SAVECUST)                        (X,COMMCUST) 001-1                 (ASLGEN'D)    42
(PR) (103)                            (GO)    (110                             )                                   (ASLGEN'D)    43
(PR) (104)        (X,TRAILID ) 001-1  (EQ)  (C,B                               )                                   (ASLGEN'D)    44
(PR) (105)                            (NS)    (108                             )                                   (ASLGEN'D)    45
(PR) (106)                            (R )  (T,SAVECUST)                        (X,COMMCUST) 001-1                 (ASLGEN'D)    46
(PR) (107)                            (GO)    (110                             )                                   (ASLGEN'D)    47
(PR) (108)                            (R )  (X,CUSTNO   ) 001-1                 (T,SAVECUST)                       (ASLGEN'D)    48
(PR) (109)                            (R )  (T,SAVECUST)                        (X,COMMCUST) 001-1                 (ASLGEN'D)    49
(PR) (110)                            (GO)    (SUB REPT__1                     )                                   (ASLGEN'D)    50
FEB 20, 2002  16.03.15                                                                                         PAGE     4
                                                       ****************************
                                                       *  REQUEST NAME - REPT__1  *
                                                       *  GENERATED REQUEST       *
                                                       ****************************
-----------------------------------------------------------------------------------------------------------------------
STMT REPORT                                 MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ         BACK BR RNT AB       SEQUENCE  LIST
TYPE  DATE          REQUESTOR ID           ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM       73 TO 80   SEQ
-----------------------------------------------------------------------------------------------------------------------
(ER)                                                                           (S)                                 (ASLGEN'D)    51
-----------------------------------------------------------------------------------------------------------------------
 ST SUM V  8  PG  PG   I SP  MAXIMUM TB COLUMN  DT  PG  STRT LIN SL RH .......SUBFILE........ C E A S R S N      C  SEQUENCE  LIST
 TY RPT S LPI WID HGT  M FRM LPP PGS TO HEADING POS POS PAGE NUM UB PN ENT    NAME   BLK/ FMT H F G T S U D      O  73 TO 80   SEQ
 MP     P              G             LT TYP POS         NUM      ML TD REC           BUF     C C S L T M S      N
-----------------------------------------------------------------------------------------------------------------------
(E1)                                                                                                               (ASLGEN'D)    52
-----------------------------------------------------------------------------------------------------------------------
           SP  Q                   E  N  S  D  C  S  T  C  C  M  M  A  M  D Q PCT/RAT      P                PARTIAL  SEQUENCE  LIST
STMT SEQ  BF  L  FIELD    SEG-    N  P  E  E  T  U  O  U  N  A  I  V  O  E L  FIELD   RA- C      PICTURE    STRT NO  73 TO 80   SEQ
TYPE NO.  COL F  NAME      LVL    D  R  Q  S  L  B  T  M  T  X  N  G  D  C F  NAME    TIO T       EDIT      CHR CHR
```

Figure D-96    Updating Application: Source Code and Report Output (Page 2 of 4)

```
(R1)         (F,M4TRAN  )                                                                                                (ASLGEN'D)    53
(R1)         (X,ALL80   ) 001-1                                                                                          (ASLGEN'D)    54
STMT  SEQ                                                                                                                SEQUENCE   LIST
TYPE  NO.                                          TEXT                                                                  73 TO 80    SEQ
(T1) (101)                  (LISTING OF HEADER AND TRAILER RECORDS#                    )                                 (ASLGEN'D)    55
FEB 20, 2002  16.03.15                                                                                                       PAGE     5
                                                 ******************************
                                                 *  PROC NAME - MAIN          *
                                                 *  INPUT STREAM PROCEDURE    *
                                                 ******************************
STMT REPORT                              MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID        ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM       73 TO 80    SEQ
(ER)                                                                                  (N)                            (ASLGEN'D)    56
    MAIN: PROC TYPE NORMAL                                                                                                   57
    ;                                                                                                                        58
    ; List the each Customer Master Record after ALL Headers and                                                            59
    ;  Trailers have been organized into the structure                                                                       60
    ;                                                                                                                        61
    REPORT CUSTNO CUSTDATA ORDERNO ORDRDATA INSTNO INSTDATA,                                                                 62
      ABBREVIATED                                                                                                            63
      GROUP BY CUSTNO                                                                                                        64
      TITLE 'THE CUSTOMER MASTER HEADER AND TRAILER FILE'                                                                    65
    END REPORT                                                                                                               66
    END PROC ;MAIN                                                                                                           67
STMT SEQ  LOG CON .....OPERAND-A.....  OPER  .............OPERAND-B.............   ......RESULT......  PARTIAL FIELD  SEQUENCE   LIST
TYPE NO.  LEV CTR QLF FIELD   SEG-LVL  ATN   QLF FIELD   SEG-LVL  (OR CONSTANT)   QLF FIELD  SEG-LVL  STRT NUMB OPR  73 TO 80    SEQ
(PR) (100)                        (GO)      (SUB REPT___2                 )                                              (ASLGEN'D)    68
FEB 20, 2002  16.03.15                                                                                                       PAGE     6
                                                 ******************************
                                                 *  REQUEST NAME - REPT___2   *
                                                 *  GENERATED REQUEST         *
                                                 ******************************
STMT REPORT                              MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID        ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM       73 TO 80    SEQ
(ER)                                                                                  (S)                 (R)      (ASLGEN'D)    69
 ST SUM V  8  PG  PG   I SP  MAXIMUM TB COLUMN  DT  PG  STRT LIN SL RH .......SUBFILE........ C E A S R S N      C  SEQUENCE   LIST
 TY RPT S LPI WID HGT  M FRM LPP PGS TO HEADING POS POS PAGE NUM UB PN ENT    NAME   BLK/ FMT H F G T S U D      O  73 TO 80    SEQ
 MP     P             G             LT TYP POS         NUM      ML TD REC           BUF      C C S L T M S      N
(E1)                                                                                                                     (ASLGEN'D)    70
         SP  Q               E N S  D  C  S  T  C  C  M  M  A  M  D Q PCT/RAT     P              PARTIAL   SEQUENCE   LIST
STMT SEQ BF  L  FIELD  SEG-  N P E  E  T  U  O  U  N  A  I  V  O  D E L  FIELD   RA- C     PICTURE     STRT NO  73 TO 80    SEQ
TYPE NO. COL F  NAME     LVL D R Q  S  L  B  T  M  T  X  N  G  D  C F  NAME    TIO T      EDIT       CHR CHR
(R1)          (CUSTNO  ) 001-1          (1)                                                                      (ASLGEN'D)    71
(R1)          (CUSTDATA) 001-1                                                                                   (ASLGEN'D)    72
(R1)          (ORDERNO ) 002-2                                                                                   (ASLGEN'D)    73
(R1)          (ORDRDATA) 002-2                                                                                   (ASLGEN'D)    74
(R1)          (INSTNO  ) 003-2                                                                                   (ASLGEN'D)    75
(R1)          (INSTDATA) 003-2                                                                                   (ASLGEN'D)    76
STMT  SEQ                                                                                                                SEQUENCE   LIST
TYPE  NO.                                          TEXT                                                                  73 TO 80    SEQ
(T1) (101)                  (THE CUSTOMER MASTER HEADER AND TRAILER FILE#              )                                 (ASLGEN'D)    77
FEB 20, 2002  16.03.15                                                                                                       PAGE     7
** MK4W204  TYPE 0  NUMBER OF MESSAGES PRINTED IS    0.                                                                          *****
                    M4NEW    -       3 RECORDS OUTPUT
                    M4TRAN   -      28 RECORDS INPUT
                    M4REPO   -      70 RECORDS OUTPUT
                    M4INPUT  -      44 RECORDS INPUT
                    M4LIST   -     233 RECORDS OUTPUT
                      15 TRACKS ASSIGNED TO M4LIB    --   13 TRACKS NOT FULL. LIBRARY DIRECTORY BLOCKING FACTOR IS  800.
** MK4W209  TYPE 0  1015528 BYTES OF MAIN STORAGE UNUSED DURING DECODING PHASE                                             *****
                    1015592 BYTES OF MAIN STORAGE UNUSED DURING COMPILATION PHASE
                     565040 BYTES OF MAIN STORAGE UNUSED DURING PROCESSING PHASE
** MK4ED02  TYPE 0  REQUEST-REPT___1  REPORT-1  REQUESTOR ID-                                                              *****
                                          LISTING OF HEADER AND TRAILER RECORDS                                  PAGE 1
     M4TRAN                                  TRAN RECORD
                                           ALL 80 POSITIONS
            1  03101CUSTOMER-DATA----------03101            03101
            2  A51010ORDER-DATA--51010                      03101
```

Figure D-96 Updating Application: Source Code and Report Output (Page 3 of 4)

```
                3  B315101INSTALL-DATA-----315101                  03101                  00001
                4  03103CUSTOMER-DATA------------03103             03103
                5  A51030ORDER-DATA--51030                         03103
                6  A51031ORDER-DATA--51031                         03103
                7  A51032ORDER-DATA--51032                         03103
                8  A51033ORDER-DATA--51033                         03103
                9  B315113INSTALL-DATA-----315113                  03103                  00001
               10  B315123INSTALL-DATA-----315123                  03103                  00001
               11  B315133INSTALL-DATA-----315133                  03103                  00001
               12  B315143INSTALL-DATA-----315143                  03103                  00001
               13  B315153INSTALL-DATA-----315153                  03103                  00001
               14  03109CUSTOMER-DATA------------03109             03109
               15  A51091ORDER-DATA--51091                         03109
               16  A51092ORDER-DATA--51092                         03109
               17  A51093ORDER-DATA--51093                         03109
               18  A51094ORDER-DATA--51094                         03109
               19  A51095ORDER-DATA--51095                         03109
               20  A51096ORDER-DATA--51096                         03109
               21  A51097ORDER-DATA--51097                         03109
               22  A51098ORDER-DATA--51098                         03109
               23  A51099ORDER-DATA--51099                         03109
               24  B315913INSTALL-DATA-----315913                  03109                  00001
               25  B315923INSTALL-DATA-----315923                  03109                  00001
               26  B315933INSTALL-DATA-----315933                  03109                  00001
               27  B315943INSTALL-DATA-----315943                  03109                  00001
               28  B315953INSTALL-DATA-----315953                  03109                  00001
** MK4FT03  TYPE 0  END OF REPORT.                                                                                  *****
** MK4ED02  TYPE 0  REQUEST-REPT___2  REPORT-1  REQUESTOR ID-                                                       *****
                                          THE CUSTOMER MASTER HEADER AND TRAILER FILE                          PAGE 1
-----------------------------------------------------------------------------------------------------------------------------
  CUSTOMER            CUSTOMER            ORDER        ORDER        INSTALL          INSTALL
   NUMBER               DATA              NUMBER        DATA         NUMBER            DATA
-----------------------------------------------------------------------------------------------------------------------------
   03101    CUSTOMER-DATA------------03101  51010   ORDER-DATA--51010  315101   INSTALL-DATA-----315101
   03103    CUSTOMER-DATA------------03103  51030   ORDER-DATA--51030  315113   INSTALL-DATA-----315113
            CUSTOMER-DATA------------03103  51031   ORDER-DATA--51031  315123   INSTALL-DATA-----315123
            CUSTOMER-DATA------------03103  51032   ORDER-DATA--51032  315133   INSTALL-DATA-----315133
            CUSTOMER-DATA------------03103  51033   ORDER-DATA--51033  315143   INSTALL-DATA-----315143
            CUSTOMER-DATA------------03103      -                   -  315153   INSTALL-DATA-----315153
   03109    CUSTOMER-DATA------------03109  51091   ORDER-DATA--51091  315913   INSTALL-DATA-----315913
            CUSTOMER-DATA------------03109  51092   ORDER-DATA--51092  315923   INSTALL-DATA-----315923
            CUSTOMER-DATA------------03109  51093   ORDER-DATA--51093  315933   INSTALL-DATA-----315933
            CUSTOMER-DATA------------03109  51094   ORDER-DATA--51094  315943   INSTALL-DATA-----315943
            CUSTOMER-DATA------------03109  51095   ORDER-DATA--51095  315953   INSTALL-DATA-----315953
            CUSTOMER-DATA------------03109  51096   ORDER-DATA--51096       -                        -
            CUSTOMER-DATA------------03109  51097   ORDER-DATA--51097       -                        -
            CUSTOMER-DATA------------03109  51098   ORDER-DATA--51098       -                        -
            CUSTOMER-DATA------------03109  51099   ORDER-DATA--51099       -                        -
** MK4FT06  TYPE 1        10 FIELDS WERE PRINTED AS INVALID, MISSING OR OVERFLOW.                                   *****
** MK4FT03  TYPE 0  END OF REPORT.                                                                                  *****
** MK4W204  TYPE 0  NUMBER OF MESSAGES PRINTED IS    5.                                                             *****
                    M4REPO   -       69 RECORDS INPUT
                    M4LIST   -       60 RECORDS OUTPUT
** MK4W209  TYPE 0  1005176 BYTES OF MAIN STORAGE UNUSED DURING REPORTING PHASE                                     *****
```

Figure D-96 Updating Application: Source Code and Report Output (Page 4 of 4)

# Report Manager

## MSTR File Definition Glossary

```
                                        DETAILED GLOSSARY BY LOCATION FOR                                  GLOSSARY PAGE      1
                                           FILE DEFINITION - MSTR
DATE CATALOGED - 03/27/02 13:14:54                                                  DATE LAST UPDATED - 03/27/02 13:14:54
CATALOGED BY -                                                                      LAST UPDATED BY -
EXPIRATION DATE -                                                                   DATE LAST USED -
FILE IDENTIFICATION =                                                               RECORD FORMAT = FIXED UNBLOCKED
NUMBER OF SEGMENTS IN FILE = 1                                                      RECORD SIZE = 80
NUMBER OF FIELDS IN FILE = 3                                                        BLOCK SIZE = 80

                                         ********************************
                                         *    SEGMENT   1, LEVEL 1     *
                                         ********************************
SEGMENT OCCURS N TIMES = 1                                  KEY FIELD 1 = DIV       TYPE = C   LENGTH = 3
```

Figure D-97 MSTR File Definition Glossary

```
SEGMENT SIZE = 19
NUMBER OF FIELDS IN SEGMENT = 3
-------------------------------------------------------------------------------------------------------------------------------
FIELD   FIELD   FIELD    FIELD   FIELD   DEC    CNT FIELD  EDIT CODES     EDIT    OUTPUT   LINE  DATA BASE NAME/
NAME    TYPE    LOCATION LENGTH  RNDING  PLACES FOR SGMT   ( ) ( ) ( )   LENGTH  WIDTH    NO    *** COLUMN HEADING ***
-------------------------------------------------------------------------------------------------------------------------------
DIV     C        1        3                                              3       8
                                                                                    1    ***   DIVISION    ***
DEPT    C        5        4                                              4       10
                                                                                    1    ***   DEPARTMENT  ***
EMPL    C       10       10                                             10       10
                                                                                    1    ***   EMPLOYEE    ***
** CLSGB00  TYPE 0  END OF FILE DEFINITION GLOSSARY.                                                          *****
```

Figure D-97 MSTR File Definition Glossary

# Standard Reports Application

## Source Listing

```
MAR 27, 2002  13.14.57                                                                                        PAGE      1
                                        ******************************************
                                        * VISION:Builder   4400  (OS/390 - 14.0)  *
                                        *           COPYRIGHT (C) 2002           *
                                        * COMPUTER ASSOCIATES INTERNATIONAL, INC. *
                                        ******************************************
VISION:Builder 14.0 GA                      ENABLED FOR IBM LANGUAGE ENVIRONMENT                  BUILD STAMP = 102045,15:49:00.
                                          BUILDER MODIFICATIONS (PTFs, APARs) INSTALLED
                                                          125,200
                                              COMLIB  MODIFICATIONS (PTFs, APARs) INSTALLED
                                        201,202,203,204,205,206,207,208,209,210,212,213,214,215,216
=================================================================================================================================
=                                                                                                                               =
=                    I N S T A L L A T I O N   P A R A M E T E R S   ( M 4 P A R A M S , M A R K L I B P )                      =
=                                                                                                                               =
=  SYSTEM DELIMITER:   #          PAGE HEIGHT:       66        M4LIST WIDTH:      132       DEF WIDTH OF PAGE:  0          =
=                                                                                                                               =
=  AUTO GRAND:         N          HEADING CHAR:      -         SUBTITLE REPEAT:   N         INVALID FIELD:      *          =
=                                                                                                                               =
=  MISSING FIELD:      -          NON-EDIT FIELD:    +         PERCENT CHAR:      %         LEFT SEPARATOR:     (          =
=                                                                                                                               =
=  RIGHT SEPARATOR:    )          SINGLE SEPARATOR:  ,         SOURCE SPACING:    1         PRINT MESSAGES:     Y          =
=                                                                                                                               =
=  CONSOLE MESSAGES:   N          M4REPO BLOCKSIZE:  4,096     INPUT I/O BUFFERS: 2         OUTPUT I/O BUFFERS: 1          =
=                                                                                                                               =
=  SNGL-STEP STORAGE:  8,192      SNGL-STEP SORTSIZE: 524,288  DIGIT SELECT CHAR: 9         ZERO SUPPRESS CHAR: Z          =
=                                                                                                                               =
=  CURRENCY CHAR:      $          PLUS CHAR:         +         MINUS CHAR:        -         CHECK PROTECT CHAR: *          =
=                                                                                                                               =
=  DECIMAL CHAR:       .          GROUPING CHAR:     ,         PRIMARY PLOT CHAR: X         SECONDARY PLOT CHAR:*          =
=                                                                                                                               =
=  FIT PLOT CHAR:      .          HORIZONTAL AXIS:   _         HORIZONTAL HASH:   |         VERTICAL AXIS:     |          =
=                                                                                                                               =
=  VERTICAL HASH:      -          MINUTES/HOUR:      60        SECONDS/MINUTE:    60        TIME DELIMITER:     HH:MM:SS   =
=                                                                                                                               =
=  DATE FORMAT:      MM DD, YYYY  TODAY FORMAT+DELIM: MM/DD/YY  ISDATE DELIMITER:  YYYY-MM-DD  JULIAN DELIMITER:  YY.DDD     =
=                                                                                                                               =
=  SORT PROGRAM CODE:  2          MINCORE VALUE:     12 K      ALT M4LIST WIDTH:  132       ALT DEF W/OF PAGE:  0          =
=                                                                                                                               =
=  MAX LINES OF TRACE: 1,024      ITEM TRACKING:     0         SUPPRESS NDS REPT?: N        DEFAULT MAXGETMN:   1,024 K    =
=                                                                                                                               =
=  CONDITION CODE 1:   0          CONDITION CODE 2:  4         CONDITION CODE 3:  8         CONDITION CODE 4:   16         =
=                                                                                                                               =
=  ISAM INDEX INCORE:  N          M4LIB BLKG FACTOR: 0         M4LIB RESERVE:     0         M4LIB COMPONENT:    C4.5 B/V   =
=                                                                                                                               =
=  DEFAULT F/P AMODE:  31         M4LIB COMPRESSION?: N        M4LIB COMPRESS MIN: 507                                      =
=                                                                                                                               =
=================================================================================================================================
MAR 27, 2002  13.14.57                                                                                        PAGE      2
-------------------------------------------------------------------------------------------------------------------------------
   RUN     STMT    FILE    O N S D U T R A B S D S L  RB  R S M R S          START          END           SEQUENCE   LIST
   NAME    TYPE    NAME    L E E I P R E U U R L C S  PL  J O O F S          SEARCH         SEARCH        73 TO 80   SEQ
                           D W Q R D N P D F T M N T  TK  T P P O R
-------------------------------------------------------------------------------------------------------------------------------
(STDRPT  ) (RC) (MSTR    ) (S) (U)      (S)       (T,G)         (R)                                    (ASLGEN'D)    1
    CONTROL NAME STDRPT TERM LISTGEN                                                                                 2
    FILE MASTER INPUT MSTR KEYS NONE                                                                             3
    FILE REPORT                                                                                                  4
                                                                                                                 5
-------------------------------------------------------------------------------------------------------------------------------
```

Figure D-98 Standard Reports Application – Source Listing (Page 1 of 3)

```
  RUN     STMT  LOGICAL  LBL  DTF/DD     KEY       KEY       KEY      QUALIFIER  USER I/O  ARRAY       PASSWORD/  SEQUENCE   LIST
  NAME    TYPE  FILENAME       NAME    FIELD-1   FIELD-2   FIELD-3   ICF CRD DBI  MODULE   QUAL        AUTH.-ID   73 TO 80    SEQ
----------------------------------------------------------------------------------------------------------------------------------
(STDRPT  )  (RF)  (MSTR    ) (S) (M4OLD   )                                                                     (ASLGEN'D)     6

==================================================================================================================================
=                                                                                                                                =
=                                                   F I L E    S U M M A R Y                                                     =
=                                                                                                                                =
= JOBNAME: KREJ004X                                STEPNAME: STDRPT                                                              =
=                                                                                                                                =
=  FILE NAME: M4OLD                                DSNAME: SYS02086.T131451.RA000.KREJ004X.M4OLD.H02                             =
=  FILE NAME: M4LIB                                DSNAME: SYS02086.T131451.RA000.KREJ004X.M4LIB.H02                             =
=                                                                                                                                =
==================================================================================================================================
MAR 27, 2002  13.14.57                                                                                                  PAGE     3
                                                     *****************************
                                                     *  PROC NAME - RPT1         *
                                                     *  INPUT STREAM PROCEDURE   *
                                                     *****************************
----------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                                 MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ        BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE         REQUESTOR ID            ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM      73 TO 80    SEQ
----------------------------------------------------------------------------------------------------------------------------------
(ER) (TODAY )                                                             (N)                                  (ASLGEN'D)     7
     RPT1:  REPORT DIV DEPT EMPL                                                                                                 8
          ORDER BY DIV DEPT EMPL                                                                                                 9
          TITLE 'REPORT 1 - DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 1'                                                          10
          FORMAT HEIGHT 22 WIDTH 75                                                                                             11
          XREP  'NOSPACELINES'                                                                                                  12
     END REPORT                                                                                                                 13
                                                                                                                                14
----------------------------------------------------------------------------------------------------------------------------------
 ST SUM V  8  PG  PG   I SP  MAXIMUM TB COLUMN  DT  PG  STRT LIN SL RH .......SUBFILE........ C E A S R S N       C  SEQUENCE  LIST
 TY RPT S LPI WID HGT  M FRM LPP PGS TO HEADING POS POS PAGE NUM UB PN ENT     NAME  BLK/ FMT H F G T S U D       O  73 TO 80   SEQ
 MP     P              G             LT TYP POS         NUM      ML TD REC           BUF     C C S L T M S       N
----------------------------------------------------------------------------------------------------------------------------------
(E1)          (075,022)                                                                                        (ASLGEN'D)    15
----------------------------------------------------------------------------------------------------------------------------------
STMT  SEQ                                                                                                      SEQUENCE   LIST
TYPE  NO.                                                 TEXT                                                 73 TO 80    SEQ
----------------------------------------------------------------------------------------------------------------------------------
(P1)  (000)                  (MARKXREP#                                               )                        (ASLGEN'D)    16
(P1)  (001)                  (NOSPACELINES#                                           )                        (ASLGEN'D)    17
----------------------------------------------------------------------------------------------------------------------------------
      SP  Q                E  N  S  D  C  S  T  C  C  M  M  A  M  D Q PCT/RAT     P                    PARTIAL  SEQUENCE   LIST
STMT SEQ  BF  L  FIELD   SEG-   N  P  E  E  T  U  O  U  N  A  I  V  O  E L  FIELD   RA- C     PICTURE      STRT NO  73 TO 80    SEQ
TYPE NO.  COL F  NAME     LVL  D  R  Q  S  L  B  T  M  T  X  N  G  D  C F  NAME    TIO T      EDIT        CHR CHR
----------------------------------------------------------------------------------------------------------------------------------
(R1)          (DIV     ) 001-1      (1)                                                                        (ASLGEN'D)    18
(R1)          (DEPT    ) 001-1      (2)                                                                        (ASLGEN'D)    19
(R1)          (EMPL    ) 001-1      (3)                                                                        (ASLGEN'D)    20
----------------------------------------------------------------------------------------------------------------------------------
STMT  SEQ                                                                                                      SEQUENCE   LIST
TYPE  NO.                                                 TEXT                                                 73 TO 80    SEQ
----------------------------------------------------------------------------------------------------------------------------------
(T1)  (101)                  (REPORT 1 - DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 1#      )                     (ASLGEN'D)    21
MAR 27, 2002  13.14.57                                                                                            PAGE     4
                                                     *****************************
                                                     *  PROC NAME - RPT1A        *
                                                     *  INPUT STREAM PROCEDURE   *
                                                     *****************************
----------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                                 MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ        BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE         REQUESTOR ID            ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM      73 TO 80    SEQ
----------------------------------------------------------------------------------------------------------------------------------
(ER) (TODAY )                                                             (N)                                  (ASLGEN'D)    22
     RPT1A: REPORT DIV DEPT EMPL                                                                                                23
          ORDER BY DIV DEPT EMPL                                                                                                24
          TITLE 'REPORT 1A- DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 1A'                                                         25
          FORMAT HEIGHT 22 WIDTH 75                                                                                             26
          XREP  'NOSPACELINES'                                                                                                  27
     END REPORT                                                                                                                 28
                                                                                                                                29
----------------------------------------------------------------------------------------------------------------------------------
 ST SUM V  8  PG  PG   I SP  MAXIMUM TB COLUMN  DT  PG  STRT LIN SL RH .......SUBFILE........ C E A S R S N       C  SEQUENCE  LIST
 TY RPT S LPI WID HGT  M FRM LPP PGS TO HEADING POS POS PAGE NUM UB PN ENT     NAME  BLK/ FMT H F G T S U D       O  73 TO 80   SEQ
 MP     P              G             LT TYP POS         NUM      ML TD REC           BUF     C C S L T M S       N
----------------------------------------------------------------------------------------------------------------------------------
(E1)          (075,022)                                                                                        (ASLGEN'D)    30
----------------------------------------------------------------------------------------------------------------------------------
STMT  SEQ                                                                                                      SEQUENCE   LIST
TYPE  NO.                                                 TEXT                                                 73 TO 80    SEQ
----------------------------------------------------------------------------------------------------------------------------------
(P1)  (000)                  (MARKXREP#                                               )                        (ASLGEN'D)    31
(P1)  (001)                  (NOSPACELINES#                                           )                        (ASLGEN'D)    32
```

Figure D-98  Standard Reports Application – Source Listing  (Page 2 of 3)

```
                SP  Q                   E  N  S  D  C  S  T  C  C  M  M  A  M  D Q PCT/RAT      P                    PARTIAL   SEQUENCE   LIST
STMT SEQ        BF  L  FIELD   SEG-     N  P  E  E  T  U  O  U  N  A  I  V  O  E L  FIELD    RA- C     PICTURE    STRT NO   73 TO 80    SEQ
TYPE NO.        COL F  NAME      LVL    D  R  Q  S  L  B  T  M  T  X  N  G  D  C F  NAME     TIO T      EDIT      CHR CHR
-------------------------------------------------------------------------------------------------------------------------------------------
(R1)             (DIV     ) 001-1       (1)                                                                                (ASLGEN'D)    33
(R1)             (DEPT    ) 001-1       (2)                                                                                (ASLGEN'D)    34
(R1)             (EMPL    ) 001-1       (3)                                                                                (ASLGEN'D)    35
-------------------------------------------------------------------------------------------------------------------------------------------
STMT  SEQ                                                                                                                   SEQUENCE   LIST
TYPE  NO.                                              TEXT                                                                 73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------------------
(T1) (101)                  (REPORT 1A- DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 1A#     )                                   (ASLGEN'D)    36
MAR 27, 2002  13.14.57                                                                                                        PAGE     5
                                                  ****************************
                                                  *  PROC NAME - RPT2        *
                                                  *  INPUT STREAM PROCEDURE  *
                                                  ****************************
-------------------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                                MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID          ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP EM       73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------------------
(ER) (TODAY )                                                                    (N)                                    (ASLGEN'D)    37
     RPT2:  REPORT DIV DEPT EMPL                                                                                                      38
          ORDER BY DIV DEPT EMPL                                                                                                      39
          TITLE 'REPORT 2- DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 2'                                                                 40
          FORMAT HEIGHT 22 WIDTH 75                                                                                                   41
          XREP  'NOSPACELINES'                                                                                                        42
     END REPORT                                                                                                                       43
-------------------------------------------------------------------------------------------------------------------------------------------
 ST SUM V  8  PG  PG   I SP  MAXIMUM TB COLUMN  DT  PG  STRT LIN SL RH .......SUBFILE..... C E A S R S N      C  SEQUENCE   LIST
 TY RPT S LPI WID HGT  M FRM LPP PGS TO HEADING POS POS PAGE NUM UB PN ENT    NAME   BLK/ FMT H F G T S U D      O  73 TO 80    SEQ
 MP     P            G            LT TYP POS        NUM      ML TD REC           BUF      C C S L T M S      N
-------------------------------------------------------------------------------------------------------------------------------------------
(E1)          (075,022)                                                                                                     (ASLGEN'D)    44
-------------------------------------------------------------------------------------------------------------------------------------------
STMT  SEQ                                                                                                                   SEQUENCE   LIST
TYPE  NO.                                              TEXT                                                                 73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------------------
(P1) (000)                  (MARKXREP#                                                  )                                   (ASLGEN'D)    45
(P1) (001)                  (NOSPACELINES#                                              )                                   (ASLGEN'D)    46
-------------------------------------------------------------------------------------------------------------------------------------------
                SP  Q                   E  N  S  D  C  S  T  C  C  M  M  A  M  D Q PCT/RAT      P                    PARTIAL   SEQUENCE   LIST
STMT SEQ        BF  L  FIELD   SEG-     N  P  E  E  T  U  O  U  N  A  I  V  O  E L  FIELD    RA- C     PICTURE    STRT NO   73 TO 80    SEQ
TYPE NO.        COL F  NAME      LVL    D  R  Q  S  L  B  T  M  T  X  N  G  D  C F  NAME     TIO T      EDIT      CHR CHR
-------------------------------------------------------------------------------------------------------------------------------------------
(R1)             (DIV     ) 001-1       (1)                                                                                (ASLGEN'D)    47
(R1)             (DEPT    ) 001-1       (2)                                                                                (ASLGEN'D)    48
(R1)             (EMPL    ) 001-1       (3)                                                                                (ASLGEN'D)    49
-------------------------------------------------------------------------------------------------------------------------------------------
STMT  SEQ                                                                                                                   SEQUENCE   LIST
TYPE  NO.                                              TEXT                                                                 73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------------------
(T1) (101)                  (REPORT 2- DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 2#       )                                   (ASLGEN'D)    50
MAR 27, 2002  13.14.57                                                                                                        PAGE     6
** MK4W204  TYPE 0  NUMBER OF MESSAGES PRINTED IS    0.                                                                                *****
                    M4OLD    -      56 RECORDS INPUT
                    M4REPO   -     206 RECORDS OUTPUT
                    M4INPUT  -      24 RECORDS INPUT
                    M4LIST   -     207 RECORDS OUTPUT
                       2 TRACKS ASSIGNED TO M4LIB    --    1 TRACKS NOT FULL. LIBRARY DIRECTORY BLOCKING FACTOR IS  109.
** MK4W209  TYPE 0  1017944 BYTES OF MAIN STORAGE UNUSED DURING DECODING PHASE                                                       *****
                    1019184 BYTES OF MAIN STORAGE UNUSED DURING COMPILATION PHASE
                     565248 BYTES OF MAIN STORAGE UNUSED DURING PROCESSING PHASE
```

Figure D-98 Standard Reports Application – Source Listing (Page 3 of 3)

## M4LIST DD Output and Reports

```
** MK4ED02  TYPE 0  REQUEST-RPT1      REPORT-1  REQUESTOR ID-                                                                      *****

03/27/02                                                                  PAGE 1
          REPORT 1 - DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 1
-------------------------------------------------------------------------------
  DIVISION  DEPARTMENT   EMPLOYEE
-------------------------------------------------------------------------------
    AAA       1111     ART
    AAA       1111     BOB
    AAA       1111     CARY
    AAA       1111     EDWARD
```

Figure D-99 Standard Reports Application – M4LIST DD Output and Reports (Page 1 of 5)

```
   AAA       1111     FRANK
   AAA       1111     JOHN
   AAA       1111     MALOU
   AAA       2222     ALAN
   AAA       2222     BILL
   AAA       2222     CRAIG
   AAA       2222     DAVID
03/27/02                                                    PAGE 2
         REPORT 1 - DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 1
--------------------------------------------------------------------------------
  DIVISION  DEPARTMENT   EMPLOYEE
--------------------------------------------------------------------------------
   AAA       2222     FREDA
   AAA       2222     JERRY
   AAA       2222     LYNN
   BBB       3333     GEORGE
   BBB       3333     HARRY
   BBB       3333     IAN
   BBB       3333     JILL
   BBB       3333     KARL
   BBB       3333     MIKE
   BBB       3333     NORM
   BBB       4444     OLIVER
03/27/02                                                    PAGE 3
         REPORT 1 - DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 1
--------------------------------------------------------------------------------
  DIVISION  DEPARTMENT   EMPLOYEE
--------------------------------------------------------------------------------
   BBB       4444     PAUL
   BBB       4444     QUINT
   BBB       4444     ROBERT
   BBB       4444     SAM
   BBB       4444     TOM
   BBB       4444     UGH
   CCC       5555     ALBERT
   CCC       5555     BAKER
   CCC       5555     CONNIE
   CCC       5555     DEBBIE
   CCC       5555     ELANOR
03/27/02                                                    PAGE 4
         REPORT 1 - DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 1
--------------------------------------------------------------------------------
  DIVISION  DEPARTMENT   EMPLOYEE
--------------------------------------------------------------------------------
   CCC       5555     FANNY
   CCC       5555     GREG
   CCC       6666     HOPE
   CCC       6666     IRA
   CCC       6666     JANE
   CCC       6666     KRIS
   CCC       6666     LESLIE
   CCC       6666     MARK
   CCC       6666     NICK
   DCC       5555     ALBERT
   DCC       5555     BAKER
03/27/02                                                    PAGE 5
         REPORT 1 - DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 1
--------------------------------------------------------------------------------
  DIVISION  DEPARTMENT   EMPLOYEE
--------------------------------------------------------------------------------
   DCC       5555     CONNIE
   DCC       5555     DEBBIE
   DCC       5555     ELANOR
   DCC       5555     FANNY
   DCC       5555     GREG
   DCC       6666     HOPE
   DCC       6666     IRA
   DCC       6666     JANE
   DCC       6666     KRIS
   DCC       6666     LESLIE
   DCC       6666     MARK
03/27/02                                                    PAGE 6
         REPORT 1 - DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 1
--------------------------------------------------------------------------------
  DIVISION  DEPARTMENT   EMPLOYEE
--------------------------------------------------------------------------------
   DCC       6666     NICK
** MK4FT03  TYPE 0  END OF REPORT.                                              *****

** MK4ED02  TYPE 0  REQUEST-RPT1A    REPORT-1  REQUESTOR ID-                    *****

03/27/02                                                    PAGE 1
         REPORT 1A- DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 1A
--------------------------------------------------------------------------------
  DIVISION  DEPARTMENT   EMPLOYEE
```

Figure D-99 Standard Reports Application – M4LIST DD Output and Reports (Page 2 of 5)

```
------------------------------------------------------------------------------
    AAA        1111     ART
    AAA        1111     BOB
    AAA        1111     CARY
    AAA        1111     EDWARD
    AAA        1111     FRANK
    AAA        1111     JOHN
    AAA        1111     MALOU
    AAA        2222     ALAN
    AAA        2222     BILL
    AAA        2222     CRAIG
    AAA        2222     DAVID
03/27/02                                                          PAGE 2
          REPORT 1A- DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 1A
------------------------------------------------------------------------------
  DIVISION  DEPARTMENT   EMPLOYEE
------------------------------------------------------------------------------
    AAA        2222     FREDA
    AAA        2222     JERRY
    AAA        2222     LYNN
    BBB        3333     GEORGE
    BBB        3333     HARRY
    BBB        3333     IAN
    BBB        3333     JILL
    BBB        3333     KARL
    BBB        3333     MIKE
    BBB        3333     NORM
    BBB        4444     OLIVER
03/27/02                                                          PAGE 3
          REPORT 1A- DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 1A
------------------------------------------------------------------------------
  DIVISION  DEPARTMENT   EMPLOYEE
------------------------------------------------------------------------------
    BBB        4444     PAUL
    BBB        4444     QUINT
    BBB        4444     ROBERT
    BBB        4444     SAM
    BBB        4444     TOM
    BBB        4444     UGH
    CCC        5555     ALBERT
    CCC        5555     BAKER
    CCC        5555     CONNIE
    CCC        5555     DEBBIE
    CCC        5555     ELANOR
03/27/02                                                          PAGE 4
          REPORT 1A- DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 1A
------------------------------------------------------------------------------
  DIVISION  DEPARTMENT   EMPLOYEE
------------------------------------------------------------------------------
    CCC        5555     FANNY
    CCC        5555     GREG
    CCC        6666     HOPE
    CCC        6666     IRA
    CCC        6666     JANE
    CCC        6666     KRIS
    CCC        6666     LESLIE
    CCC        6666     MARK
    CCC        6666     NICK
    DCC        5555     ALBERT
    DCC        5555     BAKER
03/27/02                                                          PAGE 5
          REPORT 1A- DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 1A
------------------------------------------------------------------------------
  DIVISION  DEPARTMENT   EMPLOYEE
------------------------------------------------------------------------------
    DCC        5555     CONNIE
    DCC        5555     DEBBIE
    DCC        5555     ELANOR
    DCC        5555     FANNY
    DCC        5555     GREG
    DCC        6666     HOPE
    DCC        6666     IRA
    DCC        6666     JANE
    DCC        6666     KRIS
    DCC        6666     LESLIE
    DCC        6666     MARK
03/27/02                                                          PAGE 6
          REPORT 1A- DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 1A
------------------------------------------------------------------------------
  DIVISION  DEPARTMENT   EMPLOYEE
------------------------------------------------------------------------------
    DCC        6666     NICK
** MK4FT03  TYPE 0  END OF REPORT.                                          *****

** MK4ED02  TYPE 0  REQUEST-RPT2      REPORT-1  REQUESTOR ID-               *****
```

Figure D-99 Standard Reports Application – M4LIST DD Output and Reports (Page 3 of 5)

```
03/27/02                                                        PAGE 1
          REPORT 2- DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 2
--------------------------------------------------------------------------------
  DIVISION  DEPARTMENT   EMPLOYEE
--------------------------------------------------------------------------------
    AAA        1111     ART
    AAA        1111     BOB
    AAA        1111     CARY
    AAA        1111     EDWARD
    AAA        1111     FRANK
    AAA        1111     JOHN
    AAA        1111     MALOU
    AAA        2222     ALAN
    AAA        2222     BILL
    AAA        2222     CRAIG
    AAA        2222     DAVID
03/27/02                                                        PAGE 2
          REPORT 2- DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 2
--------------------------------------------------------------------------------
  DIVISION  DEPARTMENT   EMPLOYEE
--------------------------------------------------------------------------------
    AAA        2222     FREDA
    AAA        2222     JERRY
    AAA        2222     LYNN
    BBB        3333     GEORGE
    BBB        3333     HARRY
    BBB        3333     IAN
    BBB        3333     JILL
    BBB        3333     KARL
    BBB        3333     MIKE
    BBB        3333     NORM
    BBB        4444     OLIVER
03/27/02                                                        PAGE 3
          REPORT 2- DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 2
--------------------------------------------------------------------------------
  DIVISION  DEPARTMENT   EMPLOYEE
--------------------------------------------------------------------------------
    BBB        4444     PAUL
    BBB        4444     QUINT
    BBB        4444     ROBERT
    BBB        4444     SAM
    BBB        4444     TOM
    BBB        4444     UGH
    CCC        5555     ALBERT
    CCC        5555     BAKER
    CCC        5555     CONNIE
    CCC        5555     DEBBIE
    CCC        5555     ELANOR
03/27/02                                                        PAGE 4
          REPORT 2- DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 2
--------------------------------------------------------------------------------
  DIVISION  DEPARTMENT   EMPLOYEE
--------------------------------------------------------------------------------
    CCC        5555     FANNY
    CCC        5555     GREG
    CCC        6666     HOPE
    CCC        6666     IRA
    CCC        6666     JANE
    CCC        6666     KRIS
    CCC        6666     LESLIE
    CCC        6666     MARK
    CCC        6666     NICK
    DCC        5555     ALBERT
    DCC        5555     BAKER
03/27/02                                                        PAGE 5
          REPORT 2- DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 2
--------------------------------------------------------------------------------
  DIVISION  DEPARTMENT   EMPLOYEE
--------------------------------------------------------------------------------
    DCC        5555     CONNIE
    DCC        5555     DEBBIE
    DCC        5555     ELANOR
    DCC        5555     FANNY
    DCC        5555     GREG
    DCC        6666     HOPE
    DCC        6666     IRA
    DCC        6666     JANE
    DCC        6666     KRIS
    DCC        6666     LESLIE
    DCC        6666     MARK
03/27/02                                                        PAGE 6
          REPORT 2- DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 2
```

Figure D-99 Standard Reports Application – M4LIST DD Output and Reports (Page 4 of 5)

```
-------------------------------------------------------------------------
  DIVISION  DEPARTMENT   EMPLOYEE
-------------------------------------------------------------------------
    DCC        6666     NICK
** MK4FT03  TYPE 0  END OF REPORT.                                                                                  *****
** MK4W204  TYPE 0  NUMBER OF MESSAGES PRINTED IS    6.                                                              *****
                    M4REPO   -      205 RECORDS INPUT
                    M4LIST   -      275 RECORDS OUTPUT
** MK4W209  TYPE 0  1005838 BYTES OF MAIN STORAGE UNUSED DURING REPORTING PHASE                                      *****
```

Figure D-99 Standard Reports Application – M4LIST DD Output and Reports (Page 5 of 5)

# COLLATE Reports Application

## Source Listing and M4LIST DD Output

```
MAR 27, 2002  13.15.01                                                                                    PAGE      1
                                        ********************************************
                                        *  VISION:Builder   4400  (OS/390 - 14.0)  *
                                        *            COPYRIGHT (C) 2002            *
                                        *  COMPUTER ASSOCIATES INTERNATIONAL, INC. *
                                        ********************************************
VISION:Builder 14.0 GA                      ENABLED FOR IBM LANGUAGE ENVIRONMENT                BUILD STAMP = 102045,15:49:00.
                                       BUILDER MODIFICATIONS (PTFs, APARs) INSTALLED
                                                        125,200
                                       COMLIB  MODIFICATIONS (PTFs, APARs) INSTALLED
                                 201,202,203,204,205,206,207,208,209,210,212,213,214,215,216
=====================================================================================================================
=                                                                                                                   =
=                   I N S T A L L A T I O N   P A R A M E T E R S   ( M 4 P A R A M S , M A R K L I B P )          =
=                                                                                                                   =
=  SYSTEM DELIMITER:   #          PAGE HEIGHT:        66         M4LIST WIDTH:       132        DEF WIDTH OF PAGE:  0   =
=                                                                                                                   =
=  AUTO GRAND:         N          HEADING CHAR:       -          SUBTITLE REPEAT:    N          INVALID FIELD:      *   =
=                                                                                                                   =
=  MISSING FIELD:      -          NON-EDIT FIELD:     +          PERCENT CHAR:       %          LEFT SEPARATOR:     (   =
=                                                                                                                   =
=  RIGHT SEPARATOR:    )          SINGLE SEPARATOR:   ,          SOURCE SPACING:     1          PRINT MESSAGES:     Y   =
=                                                                                                                   =
=  CONSOLE MESSAGES:   N          M4REPO BLOCKSIZE:   4,096      INPUT I/O BUFFERS:  2          OUTPUT I/O BUFFERS: 1   =
=                                                                                                                   =
=  SNGL-STEP STORAGE:  8,192      SNGL-STEP SORTSIZE: 524,288    DIGIT SELECT CHAR:  9          ZERO SUPPRESS CHAR: Z   =
=                                                                                                                   =
=  CURRENCY CHAR:      $          PLUS CHAR:          +          MINUS CHAR:         -          CHECK PROTECT CHAR: *   =
=                                                                                                                   =
=  DECIMAL CHAR:       .          GROUPING CHAR:      ,          PRIMARY PLOT CHAR:  X          SECONDARY PLOT CHAR:*   =
=                                                                                                                   =
=  FIT PLOT CHAR:      .          HORIZONTAL AXIS:    _          HORIZONTAL HASH:    |          VERTICAL AXIS:      |   =
=                                                                                                                   =
=  VERTICAL HASH:      -          MINUTES/HOUR:       60         SECONDS/MINUTE:     60         TIME DELIMITER:     HH:MM:SS =
=                                                                                                                   =
=  DATE FORMAT:      MMM DD, YYYY TODAY FORMAT+DELIM: MM/DD/YY   ISDATE DELIMITER:   YYYY-MM-DD JULIAN DELIMITER:   YY.DDD =
=                                                                                                                   =
=  SORT PROGRAM CODE:  2          MINCORE VALUE:      12 K       ALT M4LIST WIDTH:   132        ALT DEF W/OF PAGE:  0   =
=                                                                                                                   =
=  MAX LINES OF TRACE: 1,024      ITEM TRACKING:      0          SUPPRESS NDS REPT?: N          DEFAULT MAXGETMN:   1,024 K =
=                                                                                                                   =
=  CONDITION CODE 1:   0          CONDITION CODE 2:   4          CONDITION CODE 3:   8          CONDITION CODE 4:   16  =
=                                                                                                                   =
=  ISAM INDEX INCORE:  N          M4LIB BLKG FACTOR:  0          M4LIB RESERVE:      0          M4LIB COMPONENT:    C4.5 B/V =
=                                                                                                                   =
=  DEFAULT F/P AMODE:  31         M4LIB COMPRESSION?: N          M4LIB COMPRESS MIN: 507                                =
=                                                                                                                   =
=====================================================================================================================
MAR 27, 2002  13.15.01                                                                                    PAGE      2
-------------------------------------------------------------------------------------------------------------------------
   RUN     STMT    FILE    O N S D U T R A B S D S L  RB  R S M R S            START           END             SEQUENCE   LIST
   NAME    TYPE    NAME    L E E I P R E U U R L C S  PL  J O O F S            SEARCH          SEARCH          73 TO 80    SEQ
                           D W Q R D N P D F T M N T  TK  T P P O R
-------------------------------------------------------------------------------------------------------------------------
(COLLATE )  (RC)  (MSTR     ) (S)  (U)      (S)        (T,G)          (R)                                   (ASLGEN'D)     1
     CONTROL NAME COLLATE TERM LISTGEN                                                                                     2
     FILE MASTER INPUT MSTR KEYS NONE                                                                                      3
     FILE REPORT                                                                                                           4
     COLLATE REPORTS RPT2 RPT1 KEYLENGTH 1                                                                  (01351000)     5
                                                                                                                           6
```

Figure D-100 COLLATE Reports Application – Source Listing and M4LIST DD Output (Page 1 of 4)

```
----------------------------------------------------------------------------------------------------------------------------------
  RUN     STMT  LOGICAL  LBL  DTF/DD      KEY       KEY       KEY      QUALIFIER   USER I/O  ARRAY       PASSWORD/  SEQUENCE   LIST
  NAME    TYPE  FILENAME       NAME     FIELD-1   FIELD-2   FIELD-3   ICF CRD DBI   MODULE   QUAL        AUTH.-ID   73 TO 80    SEQ
----------------------------------------------------------------------------------------------------------------------------------
(COLLATE )  (RF)  (MSTR    ) (S) (M4OLD   )                                                                      (ASLGEN'D)     7

==================================================================================================================================
=                                                                                                                                =
=                                                    F I L E    S U M M A R Y                                                    =
=                                                                                                                                =
= JOBNAME: KREJ004X                                          STEPNAME: COLLATE                                                   =
=                                                                                                                                =
=  FILE NAME: M4OLD                                          DSNAME: SYS02086.T131451.RA000.KREJ004X.M4OLD.H02                   =
=  FILE NAME: M4LIB                                          DSNAME: SYS02086.T131451.RA000.KREJ004X.M4LIB.H02                   =
=                                                                                                                                =
==================================================================================================================================
MAR 27, 2002  13.15.01                                                                                                 PAGE     3
                                                   *****************************
                                                   *  PROC NAME - RMGR__1      *
                                                   *  INPUT STREAM PROCEDURE   *
                                                   *****************************
----------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                                 MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ         BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID          ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM       73 TO 80    SEQ
----------------------------------------------------------------------------------------------------------------------------------
(ER)                                                                              (Z)                            (ASLGEN'D)     8
     COLLATE REPORTS RPT2 RPT1 KEYLENGTH 1                                                                                      9
MAR 27, 2002  13.15.01                                                                                                 PAGE     4
                                                   *****************************
                                                   *  PROC NAME - RPT1         *
                                                   *  INPUT STREAM PROCEDURE   *
                                                   *****************************
----------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                                 MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ         BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID          ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM       73 TO 80    SEQ
----------------------------------------------------------------------------------------------------------------------------------
(ER) (TODAY )                                                                     (N)                            (ASLGEN'D)    10
     RPT1:  REPORT DIV DEPT EMPL                                                                                               11
          ORDER BY DIV DEPT EMPL                                                                                               12
          TITLE 'REPORT 1 - DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 1'                                                         13
          FORMAT HEIGHT 22 WIDTH 75                                                                                            14
          XREP  'NOSPACELINES'                                                                                                 15
     END REPORT                                                                                                                16
                                                                                                                               17
----------------------------------------------------------------------------------------------------------------------------------
 ST SUM V  8  PG  PG   I SP  MAXIMUM TB COLUMN  DT  PG  STRT LIN SL RH .......SUBFILE........ C E A S R S N      C  SEQUENCE   LIST
 TY RPT S LPI WID HGT  M FRM LPP PGS TO HEADING POS POS PAGE NUM UB PN ENT    NAME   BLK/ FMT H F G T S U D      O  73 TO 80    SEQ
 MP     P              G             LT TYP POS          NUM     ML TD REC           BUF     C C S L T M S      N
----------------------------------------------------------------------------------------------------------------------------------
(E1)         (075,022)                                                                                           (ASLGEN'D)    18
----------------------------------------------------------------------------------------------------------------------------------
STMT  SEQ                                                                                                        SEQUENCE   LIST
TYPE  NO.                                                 TEXT                                                   73 TO 80    SEQ
----------------------------------------------------------------------------------------------------------------------------------
(P1)  (000)                  (MARKXREP#                                                 )                        (ASLGEN'D)    19
(P1)  (001)                  (NOSPACELINES#                                             )                        (ASLGEN'D)    20
----------------------------------------------------------------------------------------------------------------------------------
      SP  Q             E  N  S  D  C  S  T  C  C  M  M  A  M  D Q PCT/RAT     P                    PARTIAL   SEQUENCE   LIST
STMT SEQ  BF  L  FIELD   SEG-  N  P  E  E  T  U  O  U  N  A  I  V  O  E L  FIELD   RA- C      PICTURE     STRT NO   73 TO 80    SEQ
TYPE NO.  COL F  NAME     LVL  D  R  Q  S  L  B  T  M  T  X  N  G  D  C F  NAME    TIO T       EDIT       CHR CHR
----------------------------------------------------------------------------------------------------------------------------------
(R1)          (DIV     ) 001-1       (1)                                                                         (ASLGEN'D)    21
(R1)          (DEPT    ) 001-1       (2)                                                                         (ASLGEN'D)    22
(R1)          (EMPL    ) 001-1       (3)                                                                         (ASLGEN'D)    23
----------------------------------------------------------------------------------------------------------------------------------
STMT  SEQ                                                                                                        SEQUENCE   LIST
TYPE  NO.                                                 TEXT                                                   73 TO 80    SEQ
----------------------------------------------------------------------------------------------------------------------------------
(T1)  (101)                  (REPORT 1 - DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 1#       )                      (ASLGEN'D)    24
MAR 27, 2002  13.15.01                                                                                                 PAGE     5
                                                   *****************************
                                                   *  PROC NAME - RPT1A        *
                                                   *  INPUT STREAM PROCEDURE   *
                                                   *****************************
----------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                                 MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ         BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID          ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM       73 TO 80    SEQ
----------------------------------------------------------------------------------------------------------------------------------
(ER) (TODAY )                                                                     (N)                            (ASLGEN'D)    25
     RPT1A: REPORT DIV DEPT EMPL                                                                                               26
          ORDER BY DIV DEPT EMPL                                                                                               27
          TITLE 'REPORT 1A- DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 1A'                                                        28
          FORMAT HEIGHT 22 WIDTH 75                                                                                            29
          XREP  'NOSPACELINES'                                                                                                 30
     END REPORT                                                                                                                31
```

Figure D-100  COLLATE Reports Application – Source Listing and M4LIST DD Output (Page 2 of 4)

```
                                                                                                                        32
--------------------------------------------------------------------------------------------------------------------------
 ST SUM V  8  PG  PG   I SP  MAXIMUM TB COLUMN  DT  PG  STRT LIN SL RH .......SUBFILE........ C E A S R S N     C  SEQUENCE   LIST
 TY RPT S LPI WID HGT  M FRM LPP PGS TO HEADING POS POS PAGE NUM UB PN ENT     NAME   BLK/ FMT H F G T S U D     O  73 TO 80    SEQ
 MP     P              G             LT TYP POS         NUM      ML TD REC            BUF     C C S L T M S     N
--------------------------------------------------------------------------------------------------------------------------
(E1)         (075,022)                                                                                      (ASLGEN'D)    33
--------------------------------------------------------------------------------------------------------------------------
STMT  SEQ                                                                                                    SEQUENCE   LIST
TYPE  NO.                                           TEXT                                                     73 TO 80    SEQ
--------------------------------------------------------------------------------------------------------------------------
(P1)  (000)                 (MARKXREP#                                                )                     (ASLGEN'D)    34
(P1)  (001)                 (NOSPACELINES#                                            )                     (ASLGEN'D)    35
--------------------------------------------------------------------------------------------------------------------------
          SP  Q                E  N  S  D  C  S  T  C  C  M  M  A  M  D Q PCT/RAT      P                PARTIAL  SEQUENCE   LIST
STMT SEQ  BF  L  FIELD   SEG-  N  P  E  E  T  U  O  U  N  A  I  V  O  E L  FIELD   RA- C     PICTURE    STRT NO  73 TO 80    SEQ
TYPE NO.  COL F  NAME     LVL  D  R  Q  S  L  B  T  M  T  X  N  G  D  C F  NAME    TIO T      EDIT      CHR CHR
--------------------------------------------------------------------------------------------------------------------------
(R1)           (DIV     ) 001-1       (1)                                                                   (ASLGEN'D)    36
(R1)           (DEPT    ) 001-1       (2)                                                                   (ASLGEN'D)    37
(R1)           (EMPL    ) 001-1       (3)                                                                   (ASLGEN'D)    38
--------------------------------------------------------------------------------------------------------------------------
STMT  SEQ                                                                                                    SEQUENCE   LIST
TYPE  NO.                                           TEXT                                                     73 TO 80    SEQ
--------------------------------------------------------------------------------------------------------------------------
(T1)  (101)                 (REPORT 1A- DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 1A#     )                   (ASLGEN'D)    39
MAR 27, 2002  13.15.01                                                                                          PAGE       6
                                                   *****************************
                                                   *  PROC NAME - RPT2         *
                                                   *  INPUT STREAM PROCEDURE   *
                                                   *****************************
--------------------------------------------------------------------------------------------------------------------------
STMT REPORT                           MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ        BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID     ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM     73 TO 80    SEQ
--------------------------------------------------------------------------------------------------------------------------
(ER) (TODAY )                                                                  (N)                          (ASLGEN'D)    40
     RPT2:  REPORT DIV DEPT EMPL                                                                                          41
          ORDER BY DIV DEPT EMPL                                                                                          42
          TITLE 'REPORT 2- DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 2'                                                     43
          FORMAT HEIGHT 22 WIDTH 75                                                                                       44
          XREP  'NOSPACELINES'                                                                                            45
     END REPORT                                                                                                           46
--------------------------------------------------------------------------------------------------------------------------
 ST SUM V  8  PG  PG   I SP  MAXIMUM TB COLUMN  DT  PG  STRT LIN SL RH .......SUBFILE........ C E A S R S N     C  SEQUENCE   LIST
 TY RPT S LPI WID HGT  M FRM LPP PGS TO HEADING POS POS PAGE NUM UB PN ENT     NAME   BLK/ FMT H F G T S U D     O  73 TO 80    SEQ
 MP     P              G             LT TYP POS         NUM      ML TD REC            BUF     C C S L T M S     N
--------------------------------------------------------------------------------------------------------------------------
(E1)         (075,022)                                                                                      (ASLGEN'D)    47
--------------------------------------------------------------------------------------------------------------------------
STMT  SEQ                                                                                                    SEQUENCE   LIST
TYPE  NO.                                           TEXT                                                     73 TO 80    SEQ
--------------------------------------------------------------------------------------------------------------------------
(P1)  (000)                 (MARKXREP#                                                )                     (ASLGEN'D)    48
(P1)  (001)                 (NOSPACELINES#                                            )                     (ASLGEN'D)    49
--------------------------------------------------------------------------------------------------------------------------
          SP  Q                E  N  S  D  C  S  T  C  C  M  M  A  M  D Q PCT/RAT      P                PARTIAL  SEQUENCE   LIST
STMT SEQ  BF  L  FIELD   SEG-  N  P  E  E  T  U  O  U  N  A  I  V  O  E L  FIELD   RA- C     PICTURE    STRT NO  73 TO 80    SEQ
TYPE NO.  COL F  NAME     LVL  D  R  Q  S  L  B  T  M  T  X  N  G  D  C F  NAME    TIO T      EDIT      CHR CHR
--------------------------------------------------------------------------------------------------------------------------
(R1)           (DIV     ) 001-1       (1)                                                                   (ASLGEN'D)    50
(R1)           (DEPT    ) 001-1       (2)                                                                   (ASLGEN'D)    51
(R1)           (EMPL    ) 001-1       (3)                                                                   (ASLGEN'D)    52
--------------------------------------------------------------------------------------------------------------------------
STMT  SEQ                                                                                                    SEQUENCE   LIST
TYPE  NO.                                           TEXT                                                     73 TO 80    SEQ
--------------------------------------------------------------------------------------------------------------------------
(T1)  (101)                 (REPORT 2- DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 2#       )                   (ASLGEN'D)    53
MAR 27, 2002  13.15.01                                                                                          PAGE       7
                                       ***  REPORT MANAGER CROSS CHECK INFORMATION  ***

--------------------------------------------------------------------------------------------------------------------------
   GROUP   STMT  GROUP   KEY      REPORT       REPORT
   NAME     ID   CODE   LENGTH     NAME        NUMBER
--------------------------------------------------------------------------------------------------------------------------
           (CO)   (A)    (  1)    (RPT2    )
           (CO)   (A)    (  1)    (RPT1    )
** MK4IZ07  TYPE 0  THERE WERE NO DECODE OR CROSS-CHECKING ERRORS FOUND WITHIN THE REPORT MANAGER STATEMENTS.             *****
MAR 27, 2002  13.15.01                                                                                          PAGE       8
** MK4W204  TYPE 0  NUMBER OF MESSAGES PRINTED IS    1.                                                                 *****
                    M4OLD   -      56 RECORDS INPUT
                    M4REPO  -     206 RECORDS OUTPUT
                    M4INPUT -      25 RECORDS INPUT
                    M4LIST  -     229 RECORDS OUTPUT
                       2 TRACKS ASSIGNED TO M4LIB    --    1 TRACKS NOT FULL. LIBRARY DIRECTORY BLOCKING FACTOR IS  109.
** MK4W209  TYPE 0  1017456 BYTES OF MAIN STORAGE UNUSED DURING DECODING PHASE                                          *****
```

Figure D-100   COLLATE Reports Application – Source Listing and M4LIST DD Output (Page 3 of 4)

```
                    1018552 BYTES OF MAIN STORAGE UNUSED DURING COMPILATION PHASE
                     565248 BYTES OF MAIN STORAGE UNUSED DURING PROCESSING PHASE
** MK4W204  TYPE 0  NUMBER OF MESSAGES PRINTED IS   18.                                                                  *****
                    M4REPO   -      277 RECORDS INPUT
                    M4LIST   -        2 RECORDS OUTPUT
** MK4W209  TYPE 0   989454 BYTES OF MAIN STORAGE UNUSED DURING REPORTING PHASE                                          *****
***** RR66-0  REPORT MANAGER OPTIONS IN EFFECT:
  - REPORT COLLATION
***** RR74-0  REPORT MANAGER WRAP UP STATISTICS:
-------------------------------------------------------------------------------------------------------------------------
      PAGES      LINES   REQUEST   FORM  REPORTS     DD/DTF     SELECTION VALUE                                MEMBER
-------------------------------------------------------------------------------------------------------------------------
         49        323  (*DEFAULT) (XX) (         ) (M4PRINT ) (                                                )(        )
```

Figure D-100 COLLATE Reports Application – Source Listing and M4LIST DD Output (Page 4 of 4)

## M4PRINT DD Output

```
** MK4ED02  TYPE 0  REQUEST-RPT2      REPORT-1  REQUESTOR ID-                                                          *****

 03/27/02                                                    PAGE 1
            REPORT 2- DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 2
 ---------------------------------------------------------------------------
   DIVISION  DEPARTMENT   EMPLOYEE
 ---------------------------------------------------------------------------
     AAA        1111     ART
     AAA        1111     BOB
     AAA        1111     CARY
     AAA        1111     EDWARD
     AAA        1111     FRANK
     AAA        1111     JOHN
     AAA        1111     MALOU
     AAA        2222     ALAN
     AAA        2222     BILL
     AAA        2222     CRAIG
     AAA        2222     DAVID
 03/27/02                                                    PAGE 2
            REPORT 2- DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 2
 ---------------------------------------------------------------------------
   DIVISION  DEPARTMENT   EMPLOYEE
 ---------------------------------------------------------------------------
     AAA        2222     FREDA
     AAA        2222     JERRY
     AAA        2222     LYNN
** MK4FT03  TYPE 0  END OF REPORT.                                                                                         *****

** MK4ED02  TYPE 0  REQUEST-RPT1      REPORT-1  REQUESTOR ID-                                                          *****

 03/27/02                                                    PAGE 1
           REPORT 1 - DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 1
 ---------------------------------------------------------------------------
   DIVISION  DEPARTMENT   EMPLOYEE
 ---------------------------------------------------------------------------
     AAA        1111     ART
     AAA        1111     BOB
     AAA        1111     CARY
     AAA        1111     EDWARD
     AAA        1111     FRANK
     AAA        1111     JOHN
     AAA        1111     MALOU
     AAA        2222     ALAN
     AAA        2222     BILL
     AAA        2222     CRAIG
     AAA        2222     DAVID
 03/27/02                                                    PAGE 2
           REPORT 1 - DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 1
 ---------------------------------------------------------------------------
   DIVISION  DEPARTMENT   EMPLOYEE
 ---------------------------------------------------------------------------
     AAA        2222     FREDA
     AAA        2222     JERRY
     AAA        2222     LYNN
** MK4FT03  TYPE 0  END OF REPORT.                                                                                         *****

** MK4ED02  TYPE 0  REQUEST-RPT2      REPORT-1  REQUESTOR ID-                                                          *****

 03/27/02                                                    PAGE 1
            REPORT 2- DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 2
 ---------------------------------------------------------------------------
   DIVISION  DEPARTMENT   EMPLOYEE
```

Figure D-101 COLLATE Reports Application – M4PRINT DD Output (Page 1 of 4)

```
------------------------------------------------------------------------------
    BBB        3333     GEORGE
    BBB        3333     HARRY
    BBB        3333     IAN
    BBB        3333     JILL
    BBB        3333     KARL
    BBB        3333     MIKE
    BBB        3333     NORM
    BBB        4444     OLIVER
    BBB        4444     PAUL
    BBB        4444     QUINT
    BBB        4444     ROBERT
03/27/02                                                        PAGE 2
          REPORT 2- DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 2
------------------------------------------------------------------------------
  DIVISION  DEPARTMENT   EMPLOYEE
------------------------------------------------------------------------------
    BBB        4444     SAM
    BBB        4444     TOM
    BBB        4444     UGH
** MK4FT03  TYPE 0  END OF REPORT.                                             *****

** MK4ED02  TYPE 0  REQUEST-RPT1      REPORT-1  REQUESTOR ID-                  *****

03/27/02                                                        PAGE 1
         REPORT 1 - DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 1
------------------------------------------------------------------------------
  DIVISION  DEPARTMENT   EMPLOYEE
------------------------------------------------------------------------------
    BBB        3333     GEORGE
    BBB        3333     HARRY
    BBB        3333     IAN
    BBB        3333     JILL
    BBB        3333     KARL
    BBB        3333     MIKE
    BBB        3333     NORM
    BBB        4444     OLIVER
    BBB        4444     PAUL
    BBB        4444     QUINT
    BBB        4444     ROBERT
03/27/02                                                        PAGE 2
         REPORT 1 - DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 1
------------------------------------------------------------------------------
  DIVISION  DEPARTMENT   EMPLOYEE
------------------------------------------------------------------------------
    BBB        4444     SAM
    BBB        4444     TOM
    BBB        4444     UGH
** MK4FT03  TYPE 0  END OF REPORT.                                             *****

** MK4ED02  TYPE 0  REQUEST-RPT2      REPORT-1  REQUESTOR ID-                  *****

03/27/02                                                        PAGE 1
          REPORT 2- DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 2
------------------------------------------------------------------------------
  DIVISION  DEPARTMENT   EMPLOYEE
------------------------------------------------------------------------------
    CCC        5555     ALBERT
    CCC        5555     BAKER
    CCC        5555     CONNIE
    CCC        5555     DEBBIE
    CCC        5555     ELANOR
    CCC        5555     FANNY
    CCC        5555     GREG
    CCC        6666     HOPE
    CCC        6666     IRA
    CCC        6666     JANE
    CCC        6666     KRIS
03/27/02                                                        PAGE 2
          REPORT 2- DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 2
------------------------------------------------------------------------------
  DIVISION  DEPARTMENT   EMPLOYEE
------------------------------------------------------------------------------
    CCC        6666     LESLIE
    CCC        6666     MARK
    CCC        6666     NICK
** MK4FT03  TYPE 0  END OF REPORT.                                             *****

** MK4ED02  TYPE 0  REQUEST-RPT1      REPORT-1  REQUESTOR ID-                  *****

03/27/02                                                        PAGE 1
         REPORT 1 - DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 1
------------------------------------------------------------------------------
  DIVISION  DEPARTMENT   EMPLOYEE
------------------------------------------------------------------------------
    CCC        5555     ALBERT
```

Figure D-101 COLLATE Reports Application – M4PRINT DD Output (Page 2 of 4)

```
   CCC       5555     BAKER
   CCC       5555     CONNIE
   CCC       5555     DEBBIE
   CCC       5555     ELANOR
   CCC       5555     FANNY
   CCC       5555     GREG
   CCC       6666     HOPE
   CCC       6666     IRA
   CCC       6666     JANE
   CCC       6666     KRIS
03/27/02                                                    PAGE 2
         REPORT 1 - DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 1
------------------------------------------------------------------------
  DIVISION  DEPARTMENT   EMPLOYEE
------------------------------------------------------------------------
   CCC       6666     LESLIE
   CCC       6666     MARK
   CCC       6666     NICK
** MK4FT03  TYPE 0  END OF REPORT.                                      *****

** MK4ED02  TYPE 0  REQUEST-RPT2     REPORT-1  REQUESTOR ID-            *****

03/27/02                                                    PAGE 1
          REPORT 2- DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 2
------------------------------------------------------------------------
  DIVISION  DEPARTMENT   EMPLOYEE
------------------------------------------------------------------------
   DCC       5555     ALBERT
   DCC       5555     BAKER
   DCC       5555     CONNIE
   DCC       5555     DEBBIE
   DCC       5555     ELANOR
   DCC       5555     FANNY
   DCC       5555     GREG
   DCC       6666     HOPE
   DCC       6666     IRA
   DCC       6666     JANE
   DCC       6666     KRIS
03/27/02                                                    PAGE 2
          REPORT 2- DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 2
------------------------------------------------------------------------
  DIVISION  DEPARTMENT   EMPLOYEE
------------------------------------------------------------------------
   DCC       6666     LESLIE
   DCC       6666     MARK
   DCC       6666     NICK
** MK4FT03  TYPE 0  END OF REPORT.                                      *****

** MK4ED02  TYPE 0  REQUEST-RPT1     REPORT-1  REQUESTOR ID-            *****

03/27/02                                                    PAGE 1
         REPORT 1 - DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 1
------------------------------------------------------------------------
  DIVISION  DEPARTMENT   EMPLOYEE
------------------------------------------------------------------------
   DCC       5555     ALBERT
   DCC       5555     BAKER
   DCC       5555     CONNIE
   DCC       5555     DEBBIE
   DCC       5555     ELANOR
   DCC       5555     FANNY
   DCC       5555     GREG
   DCC       6666     HOPE
   DCC       6666     IRA
   DCC       6666     JANE
   DCC       6666     KRIS
03/27/02                                                    PAGE 2
         REPORT 1 - DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 1
------------------------------------------------------------------------
  DIVISION  DEPARTMENT   EMPLOYEE
------------------------------------------------------------------------
   DCC       6666     LESLIE
   DCC       6666     MARK
   DCC       6666     NICK
** MK4FT03  TYPE 0  END OF REPORT.                                      *****

** MK4ED02  TYPE 0  REQUEST-RPT1A    REPORT-1  REQUESTOR ID-            *****

03/27/02                                                    PAGE 1
         REPORT 1A- DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 1A
------------------------------------------------------------------------
  DIVISION  DEPARTMENT   EMPLOYEE
------------------------------------------------------------------------
   AAA       1111     ART
   AAA       1111     BOB
   AAA       1111     CARY
```

Figure D-101 COLLATE Reports Application – M4PRINT DD Output (Page 3 of 4)

```
   AAA        1111     EDWARD
   AAA        1111     FRANK
   AAA        1111     JOHN
   AAA        1111     MALOU
   AAA        2222     ALAN
   AAA        2222     BILL
   AAA        2222     CRAIG
   AAA        2222     DAVID
03/27/02                                                          PAGE 2
         REPORT 1A- DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 1A
-------------------------------------------------------------------------------
  DIVISION  DEPARTMENT   EMPLOYEE
-------------------------------------------------------------------------------
   AAA        2222     FREDA
   AAA        2222     JERRY
   AAA        2222     LYNN
   BBB        3333     GEORGE
   BBB        3333     HARRY
   BBB        3333     IAN
   BBB        3333     JILL
   BBB        3333     KARL
   BBB        3333     MIKE
   BBB        3333     NORM
   BBB        4444     OLIVER
03/27/02                                                          PAGE 3
         REPORT 1A- DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 1A
-------------------------------------------------------------------------------
  DIVISION  DEPARTMENT   EMPLOYEE
-------------------------------------------------------------------------------
   BBB        4444     PAUL
   BBB        4444     QUINT
   BBB        4444     ROBERT
   BBB        4444     SAM
   BBB        4444     TOM
   BBB        4444     UGH
   CCC        5555     ALBERT
   CCC        5555     BAKER
   CCC        5555     CONNIE
   CCC        5555     DEBBIE
   CCC        5555     ELANOR
03/27/02                                                          PAGE 4
         REPORT 1A- DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 1A
-------------------------------------------------------------------------------
  DIVISION  DEPARTMENT   EMPLOYEE
-------------------------------------------------------------------------------
   CCC        5555     FANNY
   CCC        5555     GREG
   CCC        6666     HOPE
   CCC        6666     IRA
   CCC        6666     JANE
   CCC        6666     KRIS
   CCC        6666     LESLIE
   CCC        6666     MARK
   CCC        6666     NICK
   DCC        5555     ALBERT
   DCC        5555     BAKER
03/27/02                                                          PAGE 5
         REPORT 1A- DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 1A
-------------------------------------------------------------------------------
  DIVISION  DEPARTMENT   EMPLOYEE
-------------------------------------------------------------------------------
   DCC        5555     CONNIE
   DCC        5555     DEBBIE
   DCC        5555     ELANOR
   DCC        5555     FANNY
   DCC        5555     GREG
   DCC        6666     HOPE
   DCC        6666     IRA
   DCC        6666     JANE
   DCC        6666     KRIS
   DCC        6666     LESLIE
   DCC        6666     MARK
03/27/02                                                          PAGE 6
         REPORT 1A- DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 1A
-------------------------------------------------------------------------------
  DIVISION  DEPARTMENT   EMPLOYEE
-------------------------------------------------------------------------------
   DCC        6666     NICK
** MK4FT03  TYPE 0  END OF REPORT.                                          *****
```

Figure D-101 COLLATE Reports Application – M4PRINT DD Output (Page 4 of 4)

# ROUTE Reports Application

## Source Listing and M4LIST DD Output

```
MAR 27, 2002  13.15.05                                                                                    PAGE      1

                                          ******************************************
                                          *  VISION:Builder   4400  (OS/390 - 14.0)  *
                                          *           COPYRIGHT (C) 2002             *
                                          *  COMPUTER ASSOCIATES INTERNATIONAL, INC. *
                                          ******************************************
VISION:Builder 14.0 GA                      ENABLED FOR IBM LANGUAGE ENVIRONMENT                  BUILD STAMP = 102045,15:49:00.
                                          BUILDER MODIFICATIONS (PTFs, APARs) INSTALLED
                                                        125,200
                                            COMLIB  MODIFICATIONS (PTFs, APARs) INSTALLED
                                        201,202,203,204,205,206,207,208,209,210,212,213,214,215,216

=============================================================================================================
=                                                                                                           =
=                  I N S T A L L A T I O N   P A R A M E T E R S   ( M 4 P A R A M S , M A R K L I B P )    =
=                                                                                                           =
=  SYSTEM DELIMITER:   #          PAGE HEIGHT:        66         M4LIST WIDTH:       132        DEF WIDTH OF PAGE:  0          =
=                                                                                                           =
=  AUTO GRAND:         N          HEADING CHAR:       -          SUBTITLE REPEAT:    N          INVALID FIELD:      *          =
=                                                                                                           =
=  MISSING FIELD:      -          NON-EDIT FIELD:     +          PERCENT CHAR:       %          LEFT SEPARATOR:     (          =
=                                                                                                           =
=  RIGHT SEPARATOR:    )          SINGLE SEPARATOR:   ,          SOURCE SPACING:     1          PRINT MESSAGES:     Y          =
=                                                                                                           =
=  CONSOLE MESSAGES:   N          M4REPO BLOCKSIZE:   4,096      INPUT I/O BUFFERS:  2          OUTPUT I/O BUFFERS: 1          =
=                                                                                                           =
=  SNGL-STEP STORAGE:  8,192      SNGL-STEP SORTSIZE: 524,288    DIGIT SELECT CHAR:  9          ZERO SUPPRESS CHAR: Z          =
=                                                                                                           =
=  CURRENCY CHAR:      $          PLUS CHAR:          +          MINUS CHAR:         -          CHECK PROTECT CHAR: *          =
=                                                                                                           =
=  DECIMAL CHAR:       .          GROUPING CHAR:      ,          PRIMARY PLOT CHAR:  X          SECONDARY PLOT CHAR:*          =
=                                                                                                           =
=  FIT PLOT CHAR:      .          HORIZONTAL AXIS:    _          HORIZONTAL HASH:    |          VERTICAL AXIS:      |          =
=                                                                                                           =
=  VERTICAL HASH:      -          MINUTES/HOUR:       60         SECONDS/MINUTE:     60         TIME DELIMITER:     HH:MM:SS   =
=                                                                                                           =
=  DATE FORMAT:        MM DD, YYYY TODAY FORMAT+DELIM: MM/DD/YY   ISDATE DELIMITER:   YYYY-MM-DD JULIAN DELIMITER:   YY.DDD    =
=                                                                                                           =
=  SORT PROGRAM CODE:  2          MINCORE VALUE:      12 K       ALT M4LIST WIDTH:   132        ALT DEF W/OF PAGE:  0          =
=                                                                                                           =
=  MAX LINES OF TRACE: 1,024      ITEM TRACKING:      0          SUPPRESS NDS REPT?: N          DEFAULT MAXGETMN:   1,024 K    =
=                                                                                                           =
=  CONDITION CODE 1:   0          CONDITION CODE 2:   4          CONDITION CODE 3:   8          CONDITION CODE 4:   16         =
=                                                                                                           =
=  ISAM INDEX INCORE:  N          M4LIB BLKG FACTOR:  0          M4LIB RESERVE:      0          M4LIB COMPONENT:    C4.5 B/V   =
=                                                                                                           =
=  DEFAULT F/P AMODE:  31         M4LIB COMPRESSION?: N          M4LIB COMPRESS MIN: 507                                       =
=                                                                                                           =
=============================================================================================================

MAR 27, 2002  13.15.05                                                                                    PAGE      2
-------------------------------------------------------------------------------------------------------------------------------
   RUN     STMT    FILE    O N S D U T R A B S D S L  RB  R S M R S          START          END            SEQUENCE   LIST
   NAME    TYPE    NAME    L E E I P R E U U R L C S  PL  J O O F S          SEARCH         SEARCH         73 TO 80   SEQ
                           D W Q R D N P D F T M N T  TK  T P P O R
-------------------------------------------------------------------------------------------------------------------------------
(ROUTE   ) (RC) (MSTR    ) (S) (U)     (S)       (T,G)          (R)                                      (ASLGEN'D)     1
    CONTROL NAME ROUTE TERM LISTGEN                                                                                      2
    FILE MASTER INPUT MSTR KEYS NONE                                                                                     3
    FILE REPORT                                                                                                          4
    ROUTE REPORTS RPT1 TO ATLANTA OFFICE31                                                                (01351000)     5
    ROUTE REPORTS RPT2 KEYVALUE 'A' TO DALLAS OFFICE31                                                    (01351000)     6
    ROUTE REPORTS RPT2 KEYVALUE 'B' TO ATLANTA                                                            (01351000)     7
    ROUTE REPORTS RPT2 KEYVALUE 'C' TO OFFICE31                                                           (01351000)     8
                                                                                                                         9
-------------------------------------------------------------------------------------------------------------------------------
   RUN     STMT  LOGICAL  LBL  DTF/DD     KEY      KEY      KEY      QUALIFIER   USER I/O  ARRAY      PASSWORD/  SEQUENCE   LIST
   NAME    TYPE  FILENAME       NAME     FIELD-1  FIELD-2  FIELD-3  ICF CRD DBI  MODULE    QUAL       AUTH.-ID   73 TO 80   SEQ
-------------------------------------------------------------------------------------------------------------------------------
(ROUTE   ) (RF) (MSTR    ) (S) (M4OLD   )                                                                (ASLGEN'D)     10


=============================================================================================================
=                                                                                                           =
=                                              F I L E   S U M M A R Y                                      =
=                                                                                                           =
= JOBNAME: KREJO04X                        STEPNAME: ROUTE                                                  =
=                                                                                                           =
=  FILE NAME: M4OLD                        DSNAME: SYS02086.T131451.RA000.KREJO04X.M4OLD.H02                =
=                                                                                                           =
```

Figure D-102 Source Listing and M4LIST DD Output (Page 1 of 4)

```
=  FILE NAME: M4LIB                            DSNAME: SYS02086.T131451.RA000.KREJO04X.M4LIB.H02                                    =
=                                                                                                                                    =
======================================================================================================================================
MAR 27, 2002  13.15.05                                                                                                       PAGE    3
                                                    *****************************
                                                    *  PROC NAME - RMGR__1      *
                                                    *  INPUT STREAM PROCEDURE   *
                                                    *****************************
--------------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                             MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ         BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID       ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM       73 TO 80    SEQ
--------------------------------------------------------------------------------------------------------------------------------------
(ER)                                                                           (Z)                               (ASLGEN'D)    11
     ROUTE REPORTS RPT1 TO ATLANTA OFFICE31                                                                                    12
     ROUTE REPORTS RPT2 KEYVALUE 'A' TO DALLAS OFFICE31                                                                        13
     ROUTE REPORTS RPT2 KEYVALUE 'B' TO ATLANTA                                                                                14
     ROUTE REPORTS RPT2 KEYVALUE 'C' TO OFFICE31                                                                               15
MAR 27, 2002  13.15.05                                                                                                   PAGE    4
                                                    *****************************
                                                    *  PROC NAME - RPT1         *
                                                    *  INPUT STREAM PROCEDURE   *
                                                    *****************************
--------------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                             MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ         BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID       ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM       73 TO 80    SEQ
--------------------------------------------------------------------------------------------------------------------------------------
(ER) (TODAY )                                                                  (N)                               (ASLGEN'D)    16
     RPT1:  REPORT DIV DEPT EMPL                                                                                               17
          ORDER BY DIV DEPT EMPL                                                                                               18
          TITLE 'REPORT 1 - DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 1'                                                         19
          FORMAT HEIGHT 22 WIDTH 75                                                                                            20
          XREP  'NOSPACELINES'                                                                                                 21
     END REPORT                                                                                                                22
                                                                                                                               23
--------------------------------------------------------------------------------------------------------------------------------------
 ST SUM V  8  PG  PG   I SP  MAXIMUM TB COLUMN  DT  PG  STRT LIN SL RH .......SUBFILE........ C E A S R S N      C  SEQUENCE   LIST
 TY RPT S LPI WID HGT  M FRM LPP PGS TO HEADING POS POS PAGE NUM UB PN ENT    NAME   BLK/ FMT H F G T S U D      O  73 TO 80    SEQ
 MP     P             G             LT TYP POS         NUM      ML TD REC            BUF      C C S L T M S      N
--------------------------------------------------------------------------------------------------------------------------------------
(E1)        (075,022)                                                                                            (ASLGEN'D)    24
--------------------------------------------------------------------------------------------------------------------------------------
STMT  SEQ                                                                                                        SEQUENCE   LIST
TYPE  NO.                                              TEXT                                                      73 TO 80    SEQ
--------------------------------------------------------------------------------------------------------------------------------------
(P1)  (000)                (MARKXREP#                                                 )                          (ASLGEN'D)    25
(P1)  (001)                (NOSPACELINES#                                             )                          (ASLGEN'D)    26
--------------------------------------------------------------------------------------------------------------------------------------
          SP  Q                 E  N  S  D  C  S  T  C  C  M  M  A  M  D Q PCT/RAT     P                PARTIAL  SEQUENCE   LIST
STMT SEQ  BF  L  FIELD   SEG-   N  P  E  E  T  U  O  U  N  A  I  V  O  E L  FIELD   RA- C     PICTURE   STRT NO  73 TO 80    SEQ
TYPE NO.  COL F  NAME     LVL  D  R  Q  S  L  B  T  M  T  X  N  G  D  C F  NAME    TIO T      EDIT      CHR CHR
--------------------------------------------------------------------------------------------------------------------------------------
(R1)           (DIV      ) 001-1       (1)                                                                       (ASLGEN'D)    27
(R1)           (DEPT     ) 001-1       (2)                                                                       (ASLGEN'D)    28
(R1)           (EMPL     ) 001-1       (3)                                                                       (ASLGEN'D)    29
--------------------------------------------------------------------------------------------------------------------------------------
STMT  SEQ                                                                                                        SEQUENCE   LIST
TYPE  NO.                                              TEXT                                                      73 TO 80    SEQ
--------------------------------------------------------------------------------------------------------------------------------------
(T1)  (101)                (REPORT 1 - DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 1#     )                          (ASLGEN'D)    30
MAR 27, 2002  13.15.05                                                                                               PAGE    5
                                                    *****************************
                                                    *  PROC NAME - RPT1A        *
                                                    *  INPUT STREAM PROCEDURE   *
                                                    *****************************
--------------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                             MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ         BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID       ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM       73 TO 80    SEQ
--------------------------------------------------------------------------------------------------------------------------------------
(ER) (TODAY )                                                                  (N)                               (ASLGEN'D)    31
     RPT1A: REPORT DIV DEPT EMPL                                                                                               32
          ORDER BY DIV DEPT EMPL                                                                                               33
          TITLE 'REPORT 1A- DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 1A'                                                        34
          FORMAT HEIGHT 22 WIDTH 75                                                                                            35
          XREP  'NOSPACELINES'                                                                                                 36
     END REPORT                                                                                                                37
                                                                                                                               38
--------------------------------------------------------------------------------------------------------------------------------------
 ST SUM V  8  PG  PG   I SP  MAXIMUM TB COLUMN  DT  PG  STRT LIN SL RH .......SUBFILE........ C E A S R S N      C  SEQUENCE   LIST
 TY RPT S LPI WID HGT  M FRM LPP PGS TO HEADING POS POS PAGE NUM UB PN ENT    NAME   BLK/ FMT H F G T S U D      O  73 TO 80    SEQ
 MP     P             G             LT TYP POS         NUM      ML TD REC            BUF      C C S L T M S      N
--------------------------------------------------------------------------------------------------------------------------------------
(E1)        (075,022)                                                                                            (ASLGEN'D)    39
--------------------------------------------------------------------------------------------------------------------------------------
STMT  SEQ                                                                                                        SEQUENCE   LIST
TYPE  NO.                                              TEXT                                                      73 TO 80    SEQ
--------------------------------------------------------------------------------------------------------------------------------------
```

Figure D-102    Source Listing and M4LIST DD Output  (Page 2 of 4)

```
(P1) (000)                (MARKXREP#                                                    )                    (ASLGEN'D)    40
(P1) (001)                (NOSPACELINES#                                                )                    (ASLGEN'D)    41
-------------------------------------------------------------------------------------------------------------------------------
        SP  Q               E  N  S  D  C  S  T  C  C  M  M  A  M  D Q PCT/RAT     P                PARTIAL   SEQUENCE   LIST
STMT SEQ BF  L  FIELD   SEG-  N  P  E  E  T  U  O  U  N  A  I  V  O  E L FIELD   RA- C     PICTURE    STRT NO   73 TO 80   SEQ
TYPE NO. COL F  NAME      LVL  D  R  Q  S  L  B  T  M  T  X  N  G  D  C F NAME    TIO T      EDIT      CHR CHR
-------------------------------------------------------------------------------------------------------------------------------
(R1)          (DIV     ) 001-1      (1)                                                                    (ASLGEN'D)    42
(R1)          (DEPT    ) 001-1      (2)                                                                    (ASLGEN'D)    43
(R1)          (EMPL    ) 001-1      (3)                                                                    (ASLGEN'D)    44
-------------------------------------------------------------------------------------------------------------------------------
STMT  SEQ                                                                                                   SEQUENCE   LIST
TYPE  NO.                                          TEXT                                                     73 TO 80   SEQ
-------------------------------------------------------------------------------------------------------------------------------
(T1) (101)                (REPORT 1A- DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 1A#        )                    (ASLGEN'D)    45
MAR 27, 2002  13.15.05                                                                                        PAGE      6
                                          ****************************
                                          *  PROC NAME - RPT2        *
                                          *  INPUT STREAM PROCEDURE  *
                                          ****************************
-------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                              MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB      SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID        ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM      73 TO 80   SEQ
-------------------------------------------------------------------------------------------------------------------------------
(ER) (TODAY )                                                                 (N)                           (ASLGEN'D)    46
     RPT2:  REPORT DIV DEPT EMPL                                                                                            47
         ORDER BY DIV DEPT EMPL                                                                                             48
         TITLE 'REPORT 2- DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 2'                                                        49
         FORMAT HEIGHT 22 WIDTH 75                                                                                          50
         XREP  'NOSPACELINES'                                                                                               51
     END REPORT                                                                                                             52
-------------------------------------------------------------------------------------------------------------------------------
 ST SUM V  8  PG  PG   I SP  MAXIMUM TB COLUMN  DT  PG  STRT LIN SL RH .......SUBFILE..... C E A S R S N      C  SEQUENCE   LIST
 TY RPT S LPI WID HGT  M FRM LPP PGS TO HEADING POS POS PAGE NUM UB PN ENT    NAME   BLK/ FMT H F G T S U D      O  73 TO 80   SEQ
 MP     P            G            LT TYP POS         NUM      ML TD REC          BUF      C C S L T M S      N
-------------------------------------------------------------------------------------------------------------------------------
(E1)         (075,022)                                                                                        (ASLGEN'D)    53
-------------------------------------------------------------------------------------------------------------------------------
STMT  SEQ                                                                                                   SEQUENCE   LIST
TYPE  NO.                                          TEXT                                                     73 TO 80   SEQ
-------------------------------------------------------------------------------------------------------------------------------
(P1) (000)                (MARKXREP#                                                    )                    (ASLGEN'D)    54
(P1) (001)                (NOSPACELINES#                                                )                    (ASLGEN'D)    55
-------------------------------------------------------------------------------------------------------------------------------
        SP  Q               E  N  S  D  C  S  T  C  C  M  M  A  M  D Q PCT/RAT     P                PARTIAL   SEQUENCE   LIST
STMT SEQ BF  L  FIELD   SEG-  N  P  E  E  T  U  O  U  N  A  I  V  O  E L FIELD   RA- C     PICTURE    STRT NO   73 TO 80   SEQ
TYPE NO. COL F  NAME      LVL  D  R  Q  S  L  B  T  M  T  X  N  G  D  C F NAME    TIO T      EDIT      CHR CHR
-------------------------------------------------------------------------------------------------------------------------------
(R1)          (DIV     ) 001-1      (1)                                                                    (ASLGEN'D)    56
(R1)          (DEPT    ) 001-1      (2)                                                                    (ASLGEN'D)    57
(R1)          (EMPL    ) 001-1      (3)                                                                    (ASLGEN'D)    58
-------------------------------------------------------------------------------------------------------------------------------
STMT  SEQ                                                                                                   SEQUENCE   LIST
TYPE  NO.                                          TEXT                                                     73 TO 80   SEQ
-------------------------------------------------------------------------------------------------------------------------------
(T1) (101)                (REPORT 2- DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 2#        )                    (ASLGEN'D)    59
MAR 27, 2002  13.15.05                                                                                        PAGE      7
                                    ***  REPORT MANAGER CROSS CHECK INFORMATION  ***


-------------------------------------------------------------------------------------------------------------------------------
  REPORT   STMT   REPORT     DESTINATION                                                  PDS
   NAME     ID    NUMBER       NAME                    KEYVALUE DATA                   MEMBER
-------------------------------------------------------------------------------------------------------------------------------
(RPT1    ) (RR)             (ATLANTA )
(RPT1    ) (RR)             (OFFICE31)
(RPT2    ) (RR)             (DALLAS  ) (A                                    )
(RPT2    ) (RR)             (OFFICE31) (A                                    )
(RPT2    ) (RR)             (ATLANTA ) (B                                    )
(RPT2    ) (RR)             (OFFICE31) (C                                    )
** MK4IZ07  TYPE 0  THERE WERE NO DECODE OR CROSS-CHECKING ERRORS FOUND WITHIN THE REPORT MANAGER STATEMENTS.                 *****
MAR 27, 2002  13.15.05                                                                                        PAGE      8
** MK4W204  TYPE 0  NUMBER OF MESSAGES PRINTED IS    1.                                                         *****
                   M4OLD    -       56 RECORDS INPUT
                   M4REPO   -      207 RECORDS OUTPUT
                   M4INPUT  -       28 RECORDS INPUT
                   M4LIST   -      239 RECORDS OUTPUT
                      2 TRACKS ASSIGNED TO M4LIB    --    1 TRACKS NOT FULL. LIBRARY DIRECTORY BLOCKING FACTOR IS  109.
** MK4W209  TYPE 0  1017072 BYTES OF MAIN STORAGE UNUSED DURING DECODING PHASE                                       *****
                   1018360 BYTES OF MAIN STORAGE UNUSED DURING COMPILATION PHASE
                    565248 BYTES OF MAIN STORAGE UNUSED DURING PROCESSING PHASE
** MK4W204  TYPE 0  NUMBER OF MESSAGES PRINTED IS   12.                                                           *****
                   M4REPO   -      241 RECORDS INPUT
                   M4LIST   -        2 RECORDS OUTPUT
** MK4W209  TYPE 0   989454 BYTES OF MAIN STORAGE UNUSED DURING REPORTING PHASE                                     *****
***** RR66-0  REPORT MANAGER OPTIONS IN EFFECT:
  - REPORT ROUTING
```

Figure D-102 Source Listing and M4LIST DD Output (Page 3 of 4)

```
  - DATA BREAKING OPTIMIZATION
***** RR74-0  REPORT MANAGER WRAP UP STATISTICS:
-------------------------------------------------------------------------------------------------------------------
      PAGES      LINES   REQUEST   FORM  REPORTS     DD/DTF     SELECTION VALUE                              MEMBER
-------------------------------------------------------------------------------------------------------------------
          9         91  (RPT1    ) (XX) (         ) (ATLANTA ) (                                          ) (        )
          9         91  (RPT1    ) (XX) (         ) (OFFICE31) (                                          ) (        )
          5         29  (RPT2    ) (XX) (         ) (DALLAS  ) (A                                         ) (        )
          5         29  (RPT2    ) (XX) (         ) (OFFICE31) (A                                         ) (        )
          5         29  (RPT2    ) (XX) (         ) (ATLANTA ) (B                                         ) (        )
          5         29  (RPT2    ) (XX) (         ) (OFFICE31) (C                                         ) (        )
         14        120  (*DEFAULT) (XX) (         ) (M4PRINT ) (                                          ) (        )
```

Figure D-102 Source Listing and M4LIST DD Output (Page 4 of 4)

## M4PRINT DD Output

```
** MK4ED02  TYPE 0  REQUEST-RPT1A     REPORT-1  REQUESTOR ID-                                                    *****

03/27/02                                                    PAGE 1
          REPORT 1A- DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 1A
-------------------------------------------------------------------------
  DIVISION  DEPARTMENT   EMPLOYEE
-------------------------------------------------------------------------
    AAA        1111     ART
    AAA        1111     BOB
    AAA        1111     CARY
    AAA        1111     EDWARD
    AAA        1111     FRANK
    AAA        1111     JOHN
    AAA        1111     MALOU
    AAA        2222     ALAN
    AAA        2222     BILL
    AAA        2222     CRAIG
    AAA        2222     DAVID
03/27/02                                                    PAGE 2
          REPORT 1A- DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 1A
-------------------------------------------------------------------------
  DIVISION  DEPARTMENT   EMPLOYEE
-------------------------------------------------------------------------
    AAA        2222     FREDA
    AAA        2222     JERRY
    AAA        2222     LYNN
    BBB        3333     GEORGE
    BBB        3333     HARRY
    BBB        3333     IAN
    BBB        3333     JILL
    BBB        3333     KARL
    BBB        3333     MIKE
    BBB        3333     NORM
    BBB        4444     OLIVER
03/27/02                                                    PAGE 3
          REPORT 1A- DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 1A
-------------------------------------------------------------------------
  DIVISION  DEPARTMENT   EMPLOYEE
-------------------------------------------------------------------------
    BBB        4444     PAUL
    BBB        4444     QUINT
    BBB        4444     ROBERT
    BBB        4444     SAM
    BBB        4444     TOM
    BBB        4444     UGH
    CCC        5555     ALBERT
    CCC        5555     BAKER
    CCC        5555     CONNIE
    CCC        5555     DEBBIE
    CCC        5555     ELANOR
03/27/02                                                    PAGE 4
          REPORT 1A- DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 1A
-------------------------------------------------------------------------
  DIVISION  DEPARTMENT   EMPLOYEE
-------------------------------------------------------------------------
    CCC        5555     FANNY
    CCC        5555     GREG
    CCC        6666     HOPE
    CCC        6666     IRA
    CCC        6666     JANE
    CCC        6666     KRIS
    CCC        6666     LESLIE
    CCC        6666     MARK
    CCC        6666     NICK
    DCC        5555     ALBERT
```

Figure D-103 ROUTE Reports Application – M4PRINT DD Output (Page 1 of 2)

```
   DCC       5555     BAKER
03/27/02                                                        PAGE 5
          REPORT 1A- DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 1A
--------------------------------------------------------------------------------
  DIVISION  DEPARTMENT   EMPLOYEE
--------------------------------------------------------------------------------
   DCC       5555     CONNIE
   DCC       5555     DEBBIE
   DCC       5555     ELANOR
   DCC       5555     FANNY
   DCC       5555     GREG
   DCC       6666     HOPE
   DCC       6666     IRA
   DCC       6666     JANE
   DCC       6666     KRIS
   DCC       6666     LESLIE
   DCC       6666     MARK
03/27/02                                                        PAGE 6
          REPORT 1A- DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 1A
--------------------------------------------------------------------------------
  DIVISION  DEPARTMENT   EMPLOYEE
--------------------------------------------------------------------------------
   DCC       6666     NICK
** MK4FT03  TYPE 0  END OF REPORT.                                                *****

** MK4ED02  TYPE 0  REQUEST-RPT2     REPORT-1  REQUESTOR ID-                      *****

03/27/02                                                        PAGE 1
          REPORT 2- DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 2
--------------------------------------------------------------------------------
  DIVISION  DEPARTMENT   EMPLOYEE
--------------------------------------------------------------------------------
   DCC       5555     ALBERT
   DCC       5555     BAKER
   DCC       5555     CONNIE
   DCC       5555     DEBBIE
   DCC       5555     ELANOR
   DCC       5555     FANNY
   DCC       5555     GREG
   DCC       6666     HOPE
   DCC       6666     IRA
   DCC       6666     JANE
   DCC       6666     KRIS
03/27/02                                                        PAGE 2
          REPORT 2- DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 2
--------------------------------------------------------------------------------
  DIVISION  DEPARTMENT   EMPLOYEE
--------------------------------------------------------------------------------
   DCC       6666     LESLIE
   DCC       6666     MARK
   DCC       6666     NICK
** MK4FT03  TYPE 0  END OF REPORT.                                                *****
```

Figure D-103 ROUTE Reports Application – M4PRINT DD Output (Page 2 of 2)

## ATLANTA DD Output

```
** MK4ED02  TYPE 0  REQUEST-RPT1     REPORT-1  REQUESTOR ID-                      *****

03/27/02                                                        PAGE 1
          REPORT 1 - DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 1
--------------------------------------------------------------------------------
  DIVISION  DEPARTMENT   EMPLOYEE
--------------------------------------------------------------------------------
   AAA       1111     ART
   AAA       1111     BOB
   AAA       1111     CARY
   AAA       1111     EDWARD
   AAA       1111     FRANK
   AAA       1111     JOHN
   AAA       1111     MALOU
   AAA       2222     ALAN
   AAA       2222     BILL
   AAA       2222     CRAIG
   AAA       2222     DAVID
03/27/02                                                        PAGE 2
          REPORT 1 - DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 1
--------------------------------------------------------------------------------
  DIVISION  DEPARTMENT   EMPLOYEE
--------------------------------------------------------------------------------
   AAA       2222     FREDA
   AAA       2222     JERRY
```

Figure D-104 ROUTE Reports Application – ATLANTA DD Output (Page 1 of 3)

```
   AAA        2222     LYNN
   BBB        3333     GEORGE
   BBB        3333     HARRY
   BBB        3333     IAN
   BBB        3333     JILL
   BBB        3333     KARL
   BBB        3333     MIKE
   BBB        3333     NORM
   BBB        4444     OLIVER
03/27/02                                                    PAGE 3
         REPORT 1 - DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 1
--------------------------------------------------------------------------------
  DIVISION  DEPARTMENT   EMPLOYEE
--------------------------------------------------------------------------------
   BBB        4444     PAUL
   BBB        4444     QUINT
   BBB        4444     ROBERT
   BBB        4444     SAM
   BBB        4444     TOM
   BBB        4444     UGH
   CCC        5555     ALBERT
   CCC        5555     BAKER
   CCC        5555     CONNIE
   CCC        5555     DEBBIE
   CCC        5555     ELANOR
03/27/02                                                    PAGE 4
         REPORT 1 - DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 1
--------------------------------------------------------------------------------
  DIVISION  DEPARTMENT   EMPLOYEE
--------------------------------------------------------------------------------
   CCC        5555     FANNY
   CCC        5555     GREG
   CCC        6666     HOPE
   CCC        6666     IRA
   CCC        6666     JANE
   CCC        6666     KRIS
   CCC        6666     LESLIE
   CCC        6666     MARK
   CCC        6666     NICK
   DCC        5555     ALBERT
   DCC        5555     BAKER
03/27/02                                                    PAGE 5
         REPORT 1 - DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 1
--------------------------------------------------------------------------------
  DIVISION  DEPARTMENT   EMPLOYEE
--------------------------------------------------------------------------------
   DCC        5555     CONNIE
   DCC        5555     DEBBIE
   DCC        5555     ELANOR
   DCC        5555     FANNY
   DCC        5555     GREG
   DCC        6666     HOPE
   DCC        6666     IRA
   DCC        6666     JANE
   DCC        6666     KRIS
   DCC        6666     LESLIE
   DCC        6666     MARK
03/27/02                                                    PAGE 6
         REPORT 1 - DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 1
--------------------------------------------------------------------------------
  DIVISION  DEPARTMENT   EMPLOYEE
--------------------------------------------------------------------------------
   DCC        6666     NICK
** MK4FT03  TYPE 0  END OF REPORT.                                                *****

** MK4ED02  TYPE 0  REQUEST-RPT2      REPORT-1  REQUESTOR ID-                     *****

03/27/02                                                    PAGE 1
          REPORT 2- DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 2
--------------------------------------------------------------------------------
  DIVISION  DEPARTMENT   EMPLOYEE
--------------------------------------------------------------------------------
   BBB        3333     GEORGE
   BBB        3333     HARRY
   BBB        3333     IAN
   BBB        3333     JILL
   BBB        3333     KARL
   BBB        3333     MIKE
   BBB        3333     NORM
   BBB        4444     OLIVER
   BBB        4444     PAUL
   BBB        4444     QUINT
   BBB        4444     ROBERT
03/27/02                                                    PAGE 2
          REPORT 2- DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 2
--------------------------------------------------------------------------------
```

Figure D-104 ROUTE Reports Application – ATLANTA DD Output (Page 2 of 3)

```
  DIVISION  DEPARTMENT   EMPLOYEE
-------------------------------------------------------------------------------
    BBB        4444     SAM
    BBB        4444     TOM
    BBB        4444     UGH
** MK4FT03  TYPE 0  END OF REPORT.                                              *****
Figure D - 17-8  Report Manager Example - ROUTE Reports Application - ATLANTA DD Output

Report Manager Example - ROUTE Reports Application - DALLAS DD Output

** MK4ED02  TYPE 0  REQUEST-RPT2     REPORT-1  REQUESTOR ID-                    *****

03/27/02                                                        PAGE 1
          REPORT 2- DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 2
-------------------------------------------------------------------------------
  DIVISION  DEPARTMENT   EMPLOYEE
-------------------------------------------------------------------------------
    AAA        1111     ART
    AAA        1111     BOB
    AAA        1111     CARY
    AAA        1111     EDWARD
    AAA        1111     FRANK
    AAA        1111     JOHN
    AAA        1111     MALOU
    AAA        2222     ALAN
    AAA        2222     BILL
    AAA        2222     CRAIG
    AAA        2222     DAVID
03/27/02                                                        PAGE 2
          REPORT 2- DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 2
-------------------------------------------------------------------------------
  DIVISION  DEPARTMENT   EMPLOYEE
-------------------------------------------------------------------------------
    AAA        2222     FREDA
    AAA        2222     JERRY
    AAA        2222     LYNN
** MK4FT03  TYPE 0  END OF REPORT.                                              *****
```

Figure D-104 ROUTE Reports Application – ATLANTA DD Output (Page 3 of 3)

## DALLAS DD Output

```
Report Manager Example - ROUTE Reports Application - DALLAS DD Output

** MK4ED02  TYPE 0  REQUEST-RPT2     REPORT-1  REQUESTOR ID-                    *****

03/27/02                                                        PAGE 1
          REPORT 2- DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 2
-------------------------------------------------------------------------------
  DIVISION  DEPARTMENT   EMPLOYEE
-------------------------------------------------------------------------------
    AAA        1111     ART
    AAA        1111     BOB
    AAA        1111     CARY
    AAA        1111     EDWARD
    AAA        1111     FRANK
    AAA        1111     JOHN
    AAA        1111     MALOU
    AAA        2222     ALAN
    AAA        2222     BILL
    AAA        2222     CRAIG
    AAA        2222     DAVID
03/27/02                                                        PAGE 2
          REPORT 2- DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 2
-------------------------------------------------------------------------------
  DIVISION  DEPARTMENT   EMPLOYEE
-------------------------------------------------------------------------------
    AAA        2222     FREDA
    AAA        2222     JERRY
    AAA        2222     LYNN
** MK4FT03  TYPE 0  END OF REPORT.                                              *****
```

Figure D-105 ROUTE Reports Application – DALLAS DD Output

## OFFICE31 DD Output

```
** MK4ED02  TYPE 0  REQUEST-RPT1      REPORT-1  REQUESTOR ID-                                                                  *****

03/27/02                                                                  PAGE 1
          REPORT 1 - DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 1
-------------------------------------------------------------------------------
  DIVISION  DEPARTMENT   EMPLOYEE
-------------------------------------------------------------------------------
    AAA        1111     ART
    AAA        1111     BOB
    AAA        1111     CARY
    AAA        1111     EDWARD
    AAA        1111     FRANK
    AAA        1111     JOHN
    AAA        1111     MALOU
    AAA        2222     ALAN
    AAA        2222     BILL
    AAA        2222     CRAIG
    AAA        2222     DAVID
03/27/02                                                                  PAGE 2
          REPORT 1 - DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 1
-------------------------------------------------------------------------------
  DIVISION  DEPARTMENT   EMPLOYEE
-------------------------------------------------------------------------------
    AAA        2222     FREDA
    AAA        2222     JERRY
    AAA        2222     LYNN
    BBB        3333     GEORGE
    BBB        3333     HARRY
    BBB        3333     IAN
    BBB        3333     JILL
    BBB        3333     KARL
    BBB        3333     MIKE
    BBB        3333     NORM
    BBB        4444     OLIVER
03/27/02                                                                  PAGE 3
          REPORT 1 - DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 1
-------------------------------------------------------------------------------
  DIVISION  DEPARTMENT   EMPLOYEE
-------------------------------------------------------------------------------
    BBB        4444     PAUL
    BBB        4444     QUINT
    BBB        4444     ROBERT
    BBB        4444     SAM
    BBB        4444     TOM
    BBB        4444     UGH
    CCC        5555     ALBERT
    CCC        5555     BAKER
    CCC        5555     CONNIE
    CCC        5555     DEBBIE
    CCC        5555     ELANOR
03/27/02                                                                  PAGE 4
          REPORT 1 - DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 1
-------------------------------------------------------------------------------
  DIVISION  DEPARTMENT   EMPLOYEE
-------------------------------------------------------------------------------
    CCC        5555     FANNY
    CCC        5555     GREG
    CCC        6666     HOPE
    CCC        6666     IRA
    CCC        6666     JANE
    CCC        6666     KRIS
    CCC        6666     LESLIE
    CCC        6666     MARK
    CCC        6666     NICK
    DCC        5555     ALBERT
    DCC        5555     BAKER
03/27/02                                                                  PAGE 5
          REPORT 1 - DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 1
-------------------------------------------------------------------------------
  DIVISION  DEPARTMENT   EMPLOYEE
-------------------------------------------------------------------------------
    DCC        5555     CONNIE
    DCC        5555     DEBBIE
    DCC        5555     ELANOR
    DCC        5555     FANNY
    DCC        5555     GREG
    DCC        6666     HOPE
    DCC        6666     IRA
    DCC        6666     JANE
    DCC        6666     KRIS
    DCC        6666     LESLIE
    DCC        6666     MARK
```

Figure D-106 ROUTE Reports Application – OFFICE31 DD Output (Page 1 of 2)

```
03/27/02                                                        PAGE 6
          REPORT 1 - DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 1
--------------------------------------------------------------------------------
  DIVISION  DEPARTMENT   EMPLOYEE
--------------------------------------------------------------------------------
    DCC        6666     NICK
** MK4FT03  TYPE 0  END OF REPORT.                                              *****

** MK4ED02  TYPE 0  REQUEST-RPT2     REPORT-1  REQUESTOR ID-                    *****

03/27/02                                                        PAGE 1
          REPORT 2- DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 2
--------------------------------------------------------------------------------
  DIVISION  DEPARTMENT   EMPLOYEE
--------------------------------------------------------------------------------
    AAA        1111     ART
    AAA        1111     BOB
    AAA        1111     CARY
    AAA        1111     EDWARD
    AAA        1111     FRANK
    AAA        1111     JOHN
    AAA        1111     MALOU
    AAA        2222     ALAN
    AAA        2222     BILL
    AAA        2222     CRAIG
    AAA        2222     DAVID
03/27/02                                                        PAGE 2
          REPORT 2- DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 2
--------------------------------------------------------------------------------
  DIVISION  DEPARTMENT   EMPLOYEE
--------------------------------------------------------------------------------
    AAA        2222     FREDA
    AAA        2222     JERRY
    AAA        2222     LYNN
** MK4FT03  TYPE 0  END OF REPORT.                                              *****

** MK4ED02  TYPE 0  REQUEST-RPT2     REPORT-1  REQUESTOR ID-                    *****

03/27/02                                                        PAGE 1
          REPORT 2- DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 2
--------------------------------------------------------------------------------
  DIVISION  DEPARTMENT   EMPLOYEE
--------------------------------------------------------------------------------
    CCC        5555     ALBERT
    CCC        5555     BAKER
    CCC        5555     CONNIE
    CCC        5555     DEBBIE
    CCC        5555     ELANOR
    CCC        5555     FANNY
    CCC        5555     GREG
    CCC        6666     HOPE
    CCC        6666     IRA
    CCC        6666     JANE
    CCC        6666     KRIS
03/27/02                                                        PAGE 2
          REPORT 2- DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 2
--------------------------------------------------------------------------------
  DIVISION  DEPARTMENT   EMPLOYEE
--------------------------------------------------------------------------------
    CCC        6666     LESLIE
    CCC        6666     MARK
    CCC        6666     NICK
** MK4FT03  TYPE 0  END OF REPORT.                                              *****
```

Figure D-106 ROUTE Reports Application – OFFICE31 DD Output (Page 2 of 2)

# COLLATE and ROUTE Reports Application

## Source Listing and M4LIST DD Output

```
MAR 27, 2002  13.15.08                                                                                    PAGE      1

                                          ******************************************
                                          *  VISION:Builder   4400  (OS/390 - 14.0)  *
                                          *           COPYRIGHT (C) 2002             *
                                          *  COMPUTER ASSOCIATES INTERNATIONAL, INC. *
                                          ******************************************
VISION:Builder 14.0 GA                      ENABLED FOR IBM LANGUAGE ENVIRONMENT                  BUILD STAMP = 102045,15:49:00.
                                          BUILDER MODIFICATIONS (PTFs, APARs) INSTALLED
                                                        125,200
                                          COMLIB  MODIFICATIONS (PTFs, APARs) INSTALLED
                                        201,202,203,204,205,206,207,208,209,210,212,213,214,215,216

=============================================================================================================
=                                                                                                           =
=                  I N S T A L L A T I O N   P A R A M E T E R S   ( M 4 P A R A M S , M A R K L I B P )    =
=                                                                                                           =
=  SYSTEM DELIMITER:   #          PAGE HEIGHT:        66         M4LIST WIDTH:       132        DEF WIDTH OF PAGE:  0          =
=                                                                                                           =
=  AUTO GRAND:         N          HEADING CHAR:       -          SUBTITLE REPEAT:    N          INVALID FIELD:      *          =
=                                                                                                           =
=  MISSING FIELD:      -          NON-EDIT FIELD:     +          PERCENT CHAR:       %          LEFT SEPARATOR:     (          =
=                                                                                                           =
=  RIGHT SEPARATOR:    )          SINGLE SEPARATOR:   ,          SOURCE SPACING:     1          PRINT MESSAGES:     Y          =
=                                                                                                           =
=  CONSOLE MESSAGES:   N          M4REPO BLOCKSIZE:   4,096      INPUT I/O BUFFERS:  2          OUTPUT I/O BUFFERS: 1          =
=                                                                                                           =
=  SNGL-STEP STORAGE:  8,192      SNGL-STEP SORTSIZE: 524,288    DIGIT SELECT CHAR:  9          ZERO SUPPRESS CHAR: Z          =
=                                                                                                           =
=  CURRENCY CHAR:      $          PLUS CHAR:          +          MINUS CHAR:         -          CHECK PROTECT CHAR: *          =
=                                                                                                           =
=  DECIMAL CHAR:       .          GROUPING CHAR:      ,          PRIMARY PLOT CHAR:  X          SECONDARY PLOT CHAR:*          =
=                                                                                                           =
=  FIT PLOT CHAR:      .          HORIZONTAL AXIS:    _          HORIZONTAL HASH:    |          VERTICAL AXIS:      |          =
=                                                                                                           =
=  VERTICAL HASH:      -          MINUTES/HOUR:       60         SECONDS/MINUTE:     60         TIME DELIMITER:     HH:MM:SS   =
=                                                                                                           =
=  DATE FORMAT:      MM DD, YYYY  TODAY FORMAT+DELIM: MM/DD/YY   ISDATE DELIMITER:   YYYY-MM-DD JULIAN DELIMITER:   YY.DDD     =
=                                                                                                           =
=  SORT PROGRAM CODE:  2          MINCORE VALUE:      12 K       ALT M4LIST WIDTH:   132        ALT DEF W/OF PAGE:  0          =
=                                                                                                           =
=  MAX LINES OF TRACE: 1,024      ITEM TRACKING:      0          SUPPRESS NDS REPT?: N          DEFAULT MAXGETMN:   1,024 K    =
=                                                                                                           =
=  CONDITION CODE 1:   0          CONDITION CODE 2:   4          CONDITION CODE 3:   8          CONDITION CODE 4:   16         =
=                                                                                                           =
=  ISAM INDEX INCORE:  N          M4LIB BLKG FACTOR:  0          M4LIB RESERVE:      0          M4LIB COMPONENT:    C4.5 B/V   =
=                                                                                                           =
=  DEFAULT F/P AMODE:  31         M4LIB COMPRESSION?: N          M4LIB COMPRESS MIN: 507                                       =
=                                                                                                           =
=============================================================================================================

MAR 27, 2002  13.15.08                                                                                    PAGE      2
-------------------------------------------------------------------------------------------------------------------------
   RUN     STMT    FILE    O N S D U T R A B S D S L  RB  R S M R S          START          END            SEQUENCE   LIST
   NAME    TYPE    NAME    L E E I P R E U U R L C S  PL  J O O F S          SEARCH         SEARCH         73 TO 80    SEQ
                           D W Q R D N P D F T M N T  TK  T P P O R
-------------------------------------------------------------------------------------------------------------------------
(BOTH    ) (RC) (MSTR    ) (S) (U)     (S)       (T,G)         (R)                                      (ASLGEN'D)     1
    CONTROL NAME BOTH TERM LISTGEN                                                                                      2
    FILE MASTER INPUT MSTR KEYS NONE                                                                                    3
    FILE REPORT                                                                                                         4
    COLLATE REPORTS RPT2 RPT1 KEYLENGTH 1                                                                (01351000)     5
    ROUTE REPORTS RPT1 TO ATLANTA OFFICE31                                                               (01351000)     6
    ROUTE REPORTS RPT2 KEYVALUE 'A' TO DALLAS OFFICE31                                                   (01351000)     7
    ROUTE REPORTS RPT2 KEYVALUE 'B' TO ATLANTA                                                           (01351000)     8
    ROUTE REPORTS RPT2 KEYVALUE 'C' TO OFFICE31                                                          (01351000)     9
                                                                                                                       10
-------------------------------------------------------------------------------------------------------------------------
   RUN     STMT  LOGICAL  LBL  DTF/DD     KEY       KEY       KEY      QUALIFIER   USER I/O  ARRAY      PASSWORD/  SEQUENCE   LIST
   NAME    TYPE  FILENAME      NAME     FIELD-1   FIELD-2   FIELD-3   ICF CRD DBI   MODULE    QUAL       AUTH.-ID   73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------
(BOTH    ) (RF) (MSTR    ) (S) (M4OLD   )                                                                (ASLGEN'D)     11


=============================================================================================================
=                                                                                                           =
=                                                F I L E   S U M M A R Y                                    =
=                                                                                                           =
```

Figure D-107 COLLATE and ROUTE Reports Application – Source Listing and M4LIST DD Output (Page 1 of 4)

```
= JOBNAME: KREJO04X                          STEPNAME: BOTH                                                    =
=                                                                                                              =
=  FILE NAME: M4OLD                          DSNAME: SYS02086.T131451.RA000.KREJO04X.M4OLD.H02                 =
=  FILE NAME: M4LIB                          DSNAME: SYS02086.T131451.RA000.KREJO04X.M4LIB.H02                 =
=                                                                                                              =
================================================================================================================
MAR 27, 2002  13.15.08                                                                                PAGE      3
                                              ****************************
                                              *  PROC NAME - RMGR__1     *
                                              *  INPUT STREAM PROCEDURE  *
                                              ****************************

----------------------------------------------------------------------------------------------------------------
STMT REPORT                          MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ        BACK BR RNT AB      SEQUENCE   LIST
TYPE  DATE         REQUESTOR ID     ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM      73 TO 80    SEQ
----------------------------------------------------------------------------------------------------------------
(ER)                                                                    (Z)                          (ASLGEN'D)    12
    COLLATE REPORTS RPT2 RPT1 KEYLENGTH 1                                                                          13
    ROUTE REPORTS RPT1 TO ATLANTA OFFICE31                                                                         14
    ROUTE REPORTS RPT2 KEYVALUE 'A' TO DALLAS OFFICE31                                                             15
    ROUTE REPORTS RPT2 KEYVALUE 'B' TO ATLANTA                                                                     16
    ROUTE REPORTS RPT2 KEYVALUE 'C' TO OFFICE31                                                                    17
MAR 27, 2002  13.15.08                                                                                PAGE      4
                                              ****************************
                                              *  PROC NAME - RPT1        *
                                              *  INPUT STREAM PROCEDURE  *
                                              ****************************

----------------------------------------------------------------------------------------------------------------
STMT REPORT                          MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ        BACK BR RNT AB      SEQUENCE   LIST
TYPE  DATE         REQUESTOR ID     ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM      73 TO 80    SEQ
----------------------------------------------------------------------------------------------------------------
(ER) (TODAY )                                                           (N)                          (ASLGEN'D)    18
    RPT1:  REPORT DIV DEPT EMPL                                                                                    19
        ORDER BY DIV DEPT EMPL                                                                                     20
        TITLE 'REPORT 1 - DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 1'                                               21
        FORMAT HEIGHT 22 WIDTH 75                                                                                  22
        XREP  'NOSPACELINES'                                                                                       23
    END REPORT                                                                                                     24
                                                                                                                   25
----------------------------------------------------------------------------------------------------------------
 ST SUM V  8  PG  PG   I SP  MAXIMUM TB COLUMN  DT  PG  STRT LIN SL RH .......SUBFILE..... C E A S R S N      C  SEQUENCE   LIST
 TY RPT S LPI WID HGT  M FRM LPP PGS TO HEADING POS POS PAGE NUM UB PN ENT    NAME   BLK/ FMT H F G T S U D      O  73 TO 80    SEQ
 MP     P             G            LT TYP POS         NUM      ML TD REC           BUF      C C S L T M S      N
----------------------------------------------------------------------------------------------------------------
(E1)         (075,022)                                                                                 (ASLGEN'D)    26
----------------------------------------------------------------------------------------------------------------
STMT  SEQ                                                                                              SEQUENCE   LIST
TYPE  NO.                                        TEXT                                                  73 TO 80    SEQ
----------------------------------------------------------------------------------------------------------------
(P1)  (000)                 (MARKXREP#                                                 )               (ASLGEN'D)    27
(P1)  (001)                 (NOSPACELINES#                                             )               (ASLGEN'D)    28
----------------------------------------------------------------------------------------------------------------
        SP  Q              E N S  D  C  S  T  C  C  M  M  A  M  D Q PCT/RAT     P              PARTIAL   SEQUENCE   LIST
STMT SEQ BF  L  FIELD  SEG- N P E  E  T  U  O  U  N  A  I  V  O  E L  FIELD  RA- C     PICTURE    STRT NO  73 TO 80    SEQ
TYPE NO. COL F  NAME     LVL D R Q  S  L  B  T  M  T  X  N  G  D  C F  NAME   TIO T      EDIT      CHR CHR
----------------------------------------------------------------------------------------------------------------
(R1)          (DIV    ) 001-1     (1)                                                                 (ASLGEN'D)    29
(R1)          (DEPT   ) 001-1     (2)                                                                 (ASLGEN'D)    30
(R1)          (EMPL   ) 001-1     (3)                                                                 (ASLGEN'D)    31
----------------------------------------------------------------------------------------------------------------
STMT  SEQ                                                                                              SEQUENCE   LIST
TYPE  NO.                                        TEXT                                                  73 TO 80    SEQ
----------------------------------------------------------------------------------------------------------------
(T1)  (101)                 (REPORT 1 - DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 1#     )               (ASLGEN'D)    32
MAR 27, 2002  13.15.08                                                                                PAGE      5
                                              ****************************
                                              *  PROC NAME - RPT1A       *
                                              *  INPUT STREAM PROCEDURE  *
                                              ****************************

----------------------------------------------------------------------------------------------------------------
STMT REPORT                          MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ        BACK BR RNT AB      SEQUENCE   LIST
TYPE  DATE         REQUESTOR ID     ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM      73 TO 80    SEQ
----------------------------------------------------------------------------------------------------------------
(ER) (TODAY )                                                           (N)                          (ASLGEN'D)    33
    RPT1A: REPORT DIV DEPT EMPL                                                                                    34
        ORDER BY DIV DEPT EMPL                                                                                     35
        TITLE 'REPORT 1A- DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 1A'                                              36
        FORMAT HEIGHT 22 WIDTH 75                                                                                  37
        XREP  'NOSPACELINES'                                                                                       38
    END REPORT                                                                                                     39
                                                                                                                   40
----------------------------------------------------------------------------------------------------------------
 ST SUM V  8  PG  PG   I SP  MAXIMUM TB COLUMN  DT  PG  STRT LIN SL RH .......SUBFILE..... C E A S R S N      C  SEQUENCE   LIST
 TY RPT S LPI WID HGT  M FRM LPP PGS TO HEADING POS POS PAGE NUM UB PN ENT    NAME   BLK/ FMT H F G T S U D      O  73 TO 80    SEQ
 MP     P             G            LT TYP POS         NUM      ML TD REC           BUF      C C S L T M S      N
----------------------------------------------------------------------------------------------------------------
```

Figure D-107 COLLATE and ROUTE Reports Application – Source Listing and M4LIST DD Output (Page 2 of 4)

```
(E1)         (075,022)                                                                                          (ASLGEN'D)    41
-------------------------------------------------------------------------------------------------------------------------------
STMT  SEQ                                                                                                        SEQUENCE   LIST
TYPE  NO.                                              TEXT                                                      73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(P1)  (000)                 (MARKXREP#                                                   )                      (ASLGEN'D)    42
(P1)  (001)                 (NOSPACELINES#                                               )                      (ASLGEN'D)    43
-------------------------------------------------------------------------------------------------------------------------------
          SP  Q                      E  N  S  D  C  S  T  C  C  M  M  A  M  D Q PCT/RAT      P             PARTIAL   SEQUENCE   LIST
STMT SEQ  BF  L  FIELD   SEG-  N  P  E  E  T  U  O  U  N  A  I  V  O  E L  FIELD   RA- C     PICTURE   STRT NO   73 TO 80    SEQ
TYPE NO.  COL F  NAME      LVL  D  R  Q  S  L  B  T  M  T  X  N  G  D  C F  NAME    TIO T      EDIT     CHR CHR
-------------------------------------------------------------------------------------------------------------------------------
(R1)           (DIV     ) 001-1      (1)                                                                        (ASLGEN'D)    44
(R1)           (DEPT    ) 001-1      (2)                                                                        (ASLGEN'D)    45
(R1)           (EMPL    ) 001-1      (3)                                                                        (ASLGEN'D)    46
-------------------------------------------------------------------------------------------------------------------------------
STMT  SEQ                                                                                                        SEQUENCE   LIST
TYPE  NO.                                              TEXT                                                      73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(T1)  (101)                 (REPORT 1A- DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 1A#      )                      (ASLGEN'D)    47
MAR 27, 2002  13.15.08                                                                                              PAGE     6
                                                      *****************************
                                                      *  PROC NAME - RPT2          *
                                                      *  INPUT STREAM PROCEDURE    *
                                                      *****************************
-------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                               MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ        BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID         ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM       73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(ER) (TODAY )                                                                (N)                                (ASLGEN'D)    48
    RPT2:  REPORT DIV DEPT EMPL                                                                                               49
         ORDER BY DIV DEPT EMPL                                                                                               50
         TITLE 'REPORT 2- DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 2'                                                          51
         FORMAT HEIGHT 22 WIDTH 75                                                                                            52
         XREP  'NOSPACELINES'                                                                                                 53
    END REPORT                                                                                                                54
-------------------------------------------------------------------------------------------------------------------------------
 ST SUM V  8  PG  PG   I SP  MAXIMUM TB COLUMN  DT  PG  STRT LIN SL RH .......SUBFILE........ C E A S R S N      C  SEQUENCE   LIST
 TY RPT S LPI WID HGT  M FRM LPP PGS TO HEADING POS POS PAGE NUM UB PN ENT     NAME   BLK/ FMT H F G T S U D      O  73 TO 80    SEQ
 MP     P              G             LT TYP POS          NUM     ML TD REC            BUF     C C S L T M S      N
-------------------------------------------------------------------------------------------------------------------------------
(E1)         (075,022)                                                                                          (ASLGEN'D)    55
-------------------------------------------------------------------------------------------------------------------------------
STMT  SEQ                                                                                                        SEQUENCE   LIST
TYPE  NO.                                              TEXT                                                      73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(P1)  (000)                 (MARKXREP#                                                   )                      (ASLGEN'D)    56
(P1)  (001)                 (NOSPACELINES#                                               )                      (ASLGEN'D)    57
-------------------------------------------------------------------------------------------------------------------------------
          SP  Q                      E  N  S  D  C  S  T  C  C  M  M  A  M  D Q PCT/RAT      P             PARTIAL   SEQUENCE   LIST
STMT SEQ  BF  L  FIELD   SEG-  N  P  E  E  T  U  O  U  N  A  I  V  O  E L  FIELD   RA- C     PICTURE   STRT NO   73 TO 80    SEQ
TYPE NO.  COL F  NAME      LVL  D  R  Q  S  L  B  T  M  T  X  N  G  D  C F  NAME    TIO T      EDIT     CHR CHR
-------------------------------------------------------------------------------------------------------------------------------
(R1)           (DIV     ) 001-1      (1)                                                                        (ASLGEN'D)    58
(R1)           (DEPT    ) 001-1      (2)                                                                        (ASLGEN'D)    59
(R1)           (EMPL    ) 001-1      (3)                                                                        (ASLGEN'D)    60
-------------------------------------------------------------------------------------------------------------------------------
STMT  SEQ                                                                                                        SEQUENCE   LIST
TYPE  NO.                                              TEXT                                                      73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(T1)  (101)                 (REPORT 2- DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 2#        )                      (ASLGEN'D)    61
MAR 27, 2002  13.15.08                                                                                              PAGE     7
                                             ***  REPORT MANAGER CROSS CHECK INFORMATION  ***

-------------------------------------------------------------------------------------------------------------------------------
   GROUP   STMT  GROUP  KEY     REPORT      REPORT
    NAME    ID   CODE  LENGTH    NAME       NUMBER
-------------------------------------------------------------------------------------------------------------------------------
          (CO)   (A)   (  1)   (RPT2    )
          (CO)   (A)   (  1)   (RPT1    )
-------------------------------------------------------------------------------------------------------------------------------
  REPORT   STMT  REPORT    DESTINATION                                                PDS
   NAME     ID   NUMBER      NAME                  KEYVALUE DATA                     MEMBER
-------------------------------------------------------------------------------------------------------------------------------
(RPT1    ) (RR)             (ATLANTA )
(RPT1    ) (RR)             (OFFICE31)
(RPT2    ) (RR)             (DALLAS  )  (A                                        )
(RPT2    ) (RR)             (OFFICE31)  (A                                        )
(RPT2    ) (RR)             (ATLANTA )  (B                                        )
(RPT2    ) (RR)             (OFFICE31)  (C                                        )
** MK4IZ07  TYPE 0  THERE WERE NO DECODE OR CROSS-CHECKING ERRORS FOUND WITHIN THE REPORT MANAGER STATEMENTS.                *****
MAR 27, 2002  13.15.08                                                                                              PAGE     8
** MK4W204  TYPE 0  NUMBER OF MESSAGES PRINTED IS    1.                                                                     *****
                    M4OLD    -      56 RECORDS INPUT
                    M4REPO   -     207 RECORDS OUTPUT
```

Figure D-107    COLLATE and ROUTE Reports Application – Source Listing and M4LIST DD Output  (Page 3 of 4)

```
                    M4INPUT  -       29 RECORDS INPUT
                    M4LIST   -      247 RECORDS OUTPUT
                       2 TRACKS ASSIGNED TO M4LIB    --    1 TRACKS NOT FULL. LIBRARY DIRECTORY BLOCKING FACTOR IS  109.
** MK4W209  TYPE 0  1016880 BYTES OF MAIN STORAGE UNUSED DURING DECODING PHASE                                          *****
                    1017976 BYTES OF MAIN STORAGE UNUSED DURING COMPILATION PHASE
                     565248 BYTES OF MAIN STORAGE UNUSED DURING PROCESSING PHASE
** MK4W204  TYPE 0  NUMBER OF MESSAGES PRINTED IS   18.                                                                  *****
                    M4REPO   -      277 RECORDS INPUT
                    M4LIST   -        2 RECORDS OUTPUT
** MK4W209  TYPE 0   989454 BYTES OF MAIN STORAGE UNUSED DURING REPORTING PHASE                                          *****
***** RR66-0  REPORT MANAGER OPTIONS IN EFFECT:
  - REPORT ROUTING
  - DATA BREAKING OPTIMIZATION
  - REPORT COLLATION
***** RR74-0  REPORT MANAGER WRAP UP STATISTICS:
-----------------------------------------------------------------------------------------------------------------------
      PAGES      LINES   REQUEST   FORM  REPORTS     DD/DTF     SELECTION VALUE                              MEMBER
-----------------------------------------------------------------------------------------------------------------------
         20        116  (RPT1    ) (XX) (         ) (ATLANTA ) (                                          ) (        )
         20        116  (RPT1    ) (XX) (         ) (OFFICE31) (                                          ) (        )
          5         29  (RPT2    ) (XX) (         ) (DALLAS  ) (A                                         ) (        )
          5         29  (RPT2    ) (XX) (         ) (OFFICE31) (A                                         ) (        )
          5         29  (RPT2    ) (XX) (         ) (ATLANTA ) (B                                         ) (        )
          5         29  (RPT2    ) (XX) (         ) (OFFICE31) (C                                         ) (        )
         14        120  (*DEFAULT) (XX) (         ) (M4PRINT ) (                                          ) (        )
```

Figure D-107 COLLATE and ROUTE Reports Application – Source Listing and M4LIST DD Output (Page 4 of 4)

## M4PRINT DD Output

```
** MK4ED02  TYPE 0  REQUEST-RPT2      REPORT-1  REQUESTOR ID-                                                          *****

03/27/02                                                    PAGE 1
            REPORT 2- DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 2
-----------------------------------------------------------------------
  DIVISION  DEPARTMENT   EMPLOYEE
-----------------------------------------------------------------------
    DCC        5555     ALBERT
    DCC        5555     BAKER
    DCC        5555     CONNIE
    DCC        5555     DEBBIE
    DCC        5555     ELANOR
    DCC        5555     FANNY
    DCC        5555     GREG
    DCC        6666     HOPE
    DCC        6666     IRA
    DCC        6666     JANE
    DCC        6666     KRIS
03/27/02                                                    PAGE 2
            REPORT 2- DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 2
-----------------------------------------------------------------------
  DIVISION  DEPARTMENT   EMPLOYEE
-----------------------------------------------------------------------
    DCC        6666     LESLIE
    DCC        6666     MARK
    DCC        6666     NICK
** MK4FT03  TYPE 0  END OF REPORT.                                                                                         *****

** MK4ED02  TYPE 0  REQUEST-RPT1A     REPORT-1  REQUESTOR ID-                                                          *****

03/27/02                                                    PAGE 1
           REPORT 1A- DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 1A
-----------------------------------------------------------------------
  DIVISION  DEPARTMENT   EMPLOYEE
-----------------------------------------------------------------------
    AAA        1111     ART
    AAA        1111     BOB
    AAA        1111     CARY
    AAA        1111     EDWARD
    AAA        1111     FRANK
    AAA        1111     JOHN
    AAA        1111     MALOU
    AAA        2222     ALAN
    AAA        2222     BILL
    AAA        2222     CRAIG
    AAA        2222     DAVID
03/27/02                                                    PAGE 2
           REPORT 1A- DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 1A
-----------------------------------------------------------------------
```

Figure D-108 COLLATE and ROUTE Reports Application – M4PRINT DD Output (Page 1 of 2)

```
 DIVISION  DEPARTMENT   EMPLOYEE
-------------------------------------------------------------------------------
    AAA        2222     FREDA
    AAA        2222     JERRY
    AAA        2222     LYNN
    BBB        3333     GEORGE
    BBB        3333     HARRY
    BBB        3333     IAN
    BBB        3333     JILL
    BBB        3333     KARL
    BBB        3333     MIKE
    BBB        3333     NORM
    BBB        4444     OLIVER
03/27/02                                                          PAGE 3
          REPORT 1A- DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 1A
-------------------------------------------------------------------------------
 DIVISION  DEPARTMENT   EMPLOYEE
-------------------------------------------------------------------------------
    BBB        4444     PAUL
    BBB        4444     QUINT
    BBB        4444     ROBERT
    BBB        4444     SAM
    BBB        4444     TOM
    BBB        4444     UGH
    CCC        5555     ALBERT
    CCC        5555     BAKER
    CCC        5555     CONNIE
    CCC        5555     DEBBIE
    CCC        5555     ELANOR
03/27/02                                                          PAGE 4
          REPORT 1A- DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 1A
-------------------------------------------------------------------------------
 DIVISION  DEPARTMENT   EMPLOYEE
-------------------------------------------------------------------------------
    CCC        5555     FANNY
    CCC        5555     GREG
    CCC        6666     HOPE
    CCC        6666     IRA
    CCC        6666     JANE
    CCC        6666     KRIS
    CCC        6666     LESLIE
    CCC        6666     MARK
    CCC        6666     NICK
    DCC        5555     ALBERT
    DCC        5555     BAKER
03/27/02                                                          PAGE 5
          REPORT 1A- DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 1A
-------------------------------------------------------------------------------
 DIVISION  DEPARTMENT   EMPLOYEE
-------------------------------------------------------------------------------
    DCC        5555     CONNIE
    DCC        5555     DEBBIE
    DCC        5555     ELANOR
    DCC        5555     FANNY
    DCC        5555     GREG
    DCC        6666     HOPE
    DCC        6666     IRA
    DCC        6666     JANE
    DCC        6666     KRIS
    DCC        6666     LESLIE
    DCC        6666     MARK
03/27/02                                                          PAGE 6
          REPORT 1A- DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 1A
-------------------------------------------------------------------------------
 DIVISION  DEPARTMENT   EMPLOYEE
-------------------------------------------------------------------------------
    DCC        6666     NICK
** MK4FT03  TYPE 0  END OF REPORT.                                          *****
```

Figure D-108 COLLATE and ROUTE Reports Application – M4PRINT DD Output (Page 2 of 2)

## ATLANTA DD Output

```
** MK4ED02  TYPE 0  REQUEST-RPT1      REPORT-1  REQUESTOR ID-                                                    *****

03/27/02                                                          PAGE 1
          REPORT 1 - DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 1
-------------------------------------------------------------------------------
  DIVISION  DEPARTMENT   EMPLOYEE
-------------------------------------------------------------------------------
    AAA        1111     ART
    AAA        1111     BOB
    AAA        1111     CARY
    AAA        1111     EDWARD
    AAA        1111     FRANK
    AAA        1111     JOHN
    AAA        1111     MALOU
    AAA        2222     ALAN
    AAA        2222     BILL
    AAA        2222     CRAIG
    AAA        2222     DAVID
03/27/02                                                          PAGE 2
          REPORT 1 - DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 1
-------------------------------------------------------------------------------
  DIVISION  DEPARTMENT   EMPLOYEE
-------------------------------------------------------------------------------
    AAA        2222     FREDA
    AAA        2222     JERRY
    AAA        2222     LYNN
** MK4FT03  TYPE 0  END OF REPORT.                                                                                  *****

** MK4ED02  TYPE 0  REQUEST-RPT2      REPORT-1  REQUESTOR ID-                                                    *****

03/27/02                                                          PAGE 1
           REPORT 2- DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 2
-------------------------------------------------------------------------------
  DIVISION  DEPARTMENT   EMPLOYEE
-------------------------------------------------------------------------------
    BBB        3333     GEORGE
    BBB        3333     HARRY
    BBB        3333     IAN
    BBB        3333     JILL
    BBB        3333     KARL
    BBB        3333     MIKE
    BBB        3333     NORM
    BBB        4444     OLIVER
    BBB        4444     PAUL
    BBB        4444     QUINT
    BBB        4444     ROBERT
03/27/02                                                          PAGE 2
           REPORT 2- DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 2
-------------------------------------------------------------------------------
  DIVISION  DEPARTMENT   EMPLOYEE
-------------------------------------------------------------------------------
    BBB        4444     SAM
    BBB        4444     TOM
    BBB        4444     UGH
** MK4FT03  TYPE 0  END OF REPORT.                                                                                  *****

** MK4ED02  TYPE 0  REQUEST-RPT1      REPORT-1  REQUESTOR ID-                                                    *****

03/27/02                                                          PAGE 1
          REPORT 1 - DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 1
-------------------------------------------------------------------------------
  DIVISION  DEPARTMENT   EMPLOYEE
-------------------------------------------------------------------------------
    BBB        3333     GEORGE
    BBB        3333     HARRY
    BBB        3333     IAN
    BBB        3333     JILL
    BBB        3333     KARL
    BBB        3333     MIKE
    BBB        3333     NORM
    BBB        4444     OLIVER
    BBB        4444     PAUL
    BBB        4444     QUINT
    BBB        4444     ROBERT
03/27/02                                                          PAGE 2
          REPORT 1 - DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 1
-------------------------------------------------------------------------------
  DIVISION  DEPARTMENT   EMPLOYEE
-------------------------------------------------------------------------------
    BBB        4444     SAM
    BBB        4444     TOM
```

Figure D-109 COLLATE and ROUTE Reports Application – ATLANTA DD Output (Page 1 of 2)

```
   BBB       4444    UGH
** MK4FT03  TYPE 0  END OF REPORT.                                                    *****

** MK4ED02  TYPE 0  REQUEST-RPT1     REPORT-1  REQUESTOR ID-                         *****

03/27/02                                                          PAGE 1
         REPORT 1 - DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 1
--------------------------------------------------------------------------------
  DIVISION  DEPARTMENT   EMPLOYEE
--------------------------------------------------------------------------------
   CCC       5555    ALBERT
   CCC       5555    BAKER
   CCC       5555    CONNIE
   CCC       5555    DEBBIE
   CCC       5555    ELANOR
   CCC       5555    FANNY
   CCC       5555    GREG
   CCC       6666    HOPE
   CCC       6666    IRA
   CCC       6666    JANE
   CCC       6666    KRIS
03/27/02                                                          PAGE 2
         REPORT 1 - DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 1
--------------------------------------------------------------------------------
  DIVISION  DEPARTMENT   EMPLOYEE
--------------------------------------------------------------------------------
   CCC       6666    LESLIE
   CCC       6666    MARK
   CCC       6666    NICK
** MK4FT03  TYPE 0  END OF REPORT.                                                    *****

** MK4ED02  TYPE 0  REQUEST-RPT1     REPORT-1  REQUESTOR ID-                         *****

03/27/02                                                          PAGE 1
         REPORT 1 - DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 1
--------------------------------------------------------------------------------
  DIVISION  DEPARTMENT   EMPLOYEE
--------------------------------------------------------------------------------
   DCC       5555    ALBERT
   DCC       5555    BAKER
   DCC       5555    CONNIE
   DCC       5555    DEBBIE
   DCC       5555    ELANOR
   DCC       5555    FANNY
   DCC       5555    GREG
   DCC       6666    HOPE
   DCC       6666    IRA
   DCC       6666    JANE
   DCC       6666    KRIS
03/27/02                                                          PAGE 2
         REPORT 1 - DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 1
--------------------------------------------------------------------------------
  DIVISION  DEPARTMENT   EMPLOYEE
--------------------------------------------------------------------------------
   DCC       6666    LESLIE
   DCC       6666    MARK
   DCC       6666    NICK
** MK4FT03  TYPE 0  END OF REPORT.                                                    *****
```

Figure D-109 COLLATE and ROUTE Reports Application – ATLANTA DD Output (Page 2 of 2)

## DALLAS DD Output

```
** MK4ED02  TYPE 0  REQUEST-RPT2     REPORT-1  REQUESTOR ID-                         *****

03/27/02                                                          PAGE 1
          REPORT 2- DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 2
--------------------------------------------------------------------------------
  DIVISION  DEPARTMENT   EMPLOYEE
--------------------------------------------------------------------------------
   AAA       1111    ART
   AAA       1111    BOB
   AAA       1111    CARY
   AAA       1111    EDWARD
   AAA       1111    FRANK
   AAA       1111    JOHN
   AAA       1111    MALOU
   AAA       2222    ALAN
   AAA       2222    BILL
```

Figure D-110 COLLATE and ROUTE Reports Application – DALLAS DD Output (Page 1 of 2)

```
   AAA       2222     CRAIG
   AAA       2222     DAVID
03/27/02                                                        PAGE 2
         REPORT 2- DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 2
--------------------------------------------------------------------------------
  DIVISION  DEPARTMENT   EMPLOYEE
--------------------------------------------------------------------------------
   AAA       2222     FREDA
   AAA       2222     JERRY
   AAA       2222     LYNN
** MK4FT03  TYPE 0  END OF REPORT.                                              *****
```

Figure D-110 COLLATE and ROUTE Reports Application – DALLAS DD Output (Page 2 of 2)

## OFFICE31 DD Output

```
** MK4ED02  TYPE 0  REQUEST-RPT2     REPORT-1  REQUESTOR ID-                    *****

03/27/02                                                        PAGE 1
         REPORT 2- DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 2
--------------------------------------------------------------------------------
  DIVISION  DEPARTMENT   EMPLOYEE
--------------------------------------------------------------------------------
   AAA       1111     ART
   AAA       1111     BOB
   AAA       1111     CARY
   AAA       1111     EDWARD
   AAA       1111     FRANK
   AAA       1111     JOHN
   AAA       1111     MALOU
   AAA       2222     ALAN
   AAA       2222     BILL
   AAA       2222     CRAIG
   AAA       2222     DAVID
03/27/02                                                        PAGE 2
         REPORT 2- DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 2
--------------------------------------------------------------------------------
  DIVISION  DEPARTMENT   EMPLOYEE
--------------------------------------------------------------------------------
   AAA       2222     FREDA
   AAA       2222     JERRY
   AAA       2222     LYNN
** MK4FT03  TYPE 0  END OF REPORT.                                              *****

** MK4ED02  TYPE 0  REQUEST-RPT1     REPORT-1  REQUESTOR ID-                    *****

03/27/02                                                        PAGE 1
         REPORT 1 - DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 1
--------------------------------------------------------------------------------
  DIVISION  DEPARTMENT   EMPLOYEE
--------------------------------------------------------------------------------
   AAA       1111     ART
   AAA       1111     BOB
   AAA       1111     CARY
   AAA       1111     EDWARD
   AAA       1111     FRANK
   AAA       1111     JOHN
   AAA       1111     MALOU
   AAA       2222     ALAN
   AAA       2222     BILL
   AAA       2222     CRAIG
   AAA       2222     DAVID
03/27/02                                                        PAGE 2
         REPORT 1 - DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 1
--------------------------------------------------------------------------------
  DIVISION  DEPARTMENT   EMPLOYEE
--------------------------------------------------------------------------------
   AAA       2222     FREDA
   AAA       2222     JERRY
   AAA       2222     LYNN
** MK4FT03  TYPE 0  END OF REPORT.                                              *****

** MK4ED02  TYPE 0  REQUEST-RPT1     REPORT-1  REQUESTOR ID-                    *****

03/27/02                                                        PAGE 1
         REPORT 1 - DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 1
--------------------------------------------------------------------------------
  DIVISION  DEPARTMENT   EMPLOYEE
--------------------------------------------------------------------------------
```

Figure D-111 COLLATE and ROUTE Reports Application – OFFICE31 DD Output (Page 1 of 3)

```
   BBB       3333     GEORGE
   BBB       3333     HARRY
   BBB       3333     IAN
   BBB       3333     JILL
   BBB       3333     KARL
   BBB       3333     MIKE
   BBB       3333     NORM
   BBB       4444     OLIVER
   BBB       4444     PAUL
   BBB       4444     QUINT
   BBB       4444     ROBERT
03/27/02                                                    PAGE 2
         REPORT 1 - DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 1
------------------------------------------------------------------------
  DIVISION  DEPARTMENT   EMPLOYEE
------------------------------------------------------------------------
   BBB       4444     SAM
   BBB       4444     TOM
   BBB       4444     UGH
** MK4FT03  TYPE 0  END OF REPORT.                                      *****

** MK4ED02  TYPE 0  REQUEST-RPT2     REPORT-1  REQUESTOR ID-           *****

03/27/02                                                    PAGE 1
          REPORT 2- DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 2
------------------------------------------------------------------------
  DIVISION  DEPARTMENT   EMPLOYEE
------------------------------------------------------------------------
   CCC       5555     ALBERT
   CCC       5555     BAKER
   CCC       5555     CONNIE
   CCC       5555     DEBBIE
   CCC       5555     ELANOR
   CCC       5555     FANNY
   CCC       5555     GREG
   CCC       6666     HOPE
   CCC       6666     IRA
   CCC       6666     JANE
   CCC       6666     KRIS
03/27/02                                                    PAGE 2
          REPORT 2- DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 2
------------------------------------------------------------------------
  DIVISION  DEPARTMENT   EMPLOYEE
------------------------------------------------------------------------
   CCC       6666     LESLIE
   CCC       6666     MARK
   CCC       6666     NICK
** MK4FT03  TYPE 0  END OF REPORT.                                      *****

** MK4ED02  TYPE 0  REQUEST-RPT1     REPORT-1  REQUESTOR ID-           *****

03/27/02                                                    PAGE 1
         REPORT 1 - DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 1
------------------------------------------------------------------------
  DIVISION  DEPARTMENT   EMPLOYEE
------------------------------------------------------------------------
   CCC       5555     ALBERT
   CCC       5555     BAKER
   CCC       5555     CONNIE
   CCC       5555     DEBBIE
   CCC       5555     ELANOR
   CCC       5555     FANNY
   CCC       5555     GREG
   CCC       6666     HOPE
   CCC       6666     IRA
   CCC       6666     JANE
   CCC       6666     KRIS
03/27/02                                                    PAGE 2
         REPORT 1 - DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 1
------------------------------------------------------------------------
  DIVISION  DEPARTMENT   EMPLOYEE
------------------------------------------------------------------------
   CCC       6666     LESLIE
   CCC       6666     MARK
   CCC       6666     NICK
** MK4FT03  TYPE 0  END OF REPORT.                                      *****

** MK4ED02  TYPE 0  REQUEST-RPT1     REPORT-1  REQUESTOR ID-           *****

03/27/02                                                    PAGE 1
         REPORT 1 - DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 1
------------------------------------------------------------------------
  DIVISION  DEPARTMENT   EMPLOYEE
------------------------------------------------------------------------
   DCC       5555     ALBERT
```

Figure D-111 COLLATE and ROUTE Reports Application – OFFICE31 DD Output (Page 2 of 3)

```
    DCC        5555     BAKER
    DCC        5555     CONNIE
    DCC        5555     DEBBIE
    DCC        5555     ELANOR
    DCC        5555     FANNY
    DCC        5555     GREG
    DCC        6666     HOPE
    DCC        6666     IRA
    DCC        6666     JANE
    DCC        6666     KRIS
03/27/02                                                    PAGE 2
          REPORT 1 - DIVISION/DEPARTMENT/EMPL - ORIG RPT NO. 1
-----------------------------------------------------------------------
  DIVISION  DEPARTMENT   EMPLOYEE
-----------------------------------------------------------------------
    DCC        6666     LESLIE
    DCC        6666     MARK
    DCC        6666     NICK
** MK4FT03  TYPE 0  END OF REPORT.                                                                                                  *****
```

Figure D-111 COLLATE and ROUTE Reports Application – OFFICE31 DD Output (Page 3 of 3)

# Cataloging

## Cataloging Example

```
-----------------------------------------------------------------------------------------------------------------------------------
  RUN     STMT    FILE    O N S D U T R A B S D S L  RB  R S M R S            START            END              SEQUENCE   LIST
  NAME    TYPE    NAME    L E E I P R E U U R L C S  PL  J O O F S            SEARCH           SEARCH           73 TO 80    SEQ
                          D W Q R D N P D F T M N T  TK  T P P O R
-----------------------------------------------------------------------------------------------------------------------------------
(ASLRC   ) (RC) (SAMPLE  ) (S)          (S)          (G)          (R)                                                (ASLGEN'D)     1

    CONTROL  SYSDATE 061402   LISTGEN
    FILE MASTER INPUT SAMPLE
    FILE REPORT
    ;
    ;;ENDASL
-----------------------------------------------------------------------------------------------------------------------------------
  RUN     STMT  LOGICAL  LBL  DTF/DD      KEY       KEY       KEY      QUALIFIER   USER I/O  ARRAY       PASSWORD/  SEQUENCE   LIST
  NAME    TYPE  FILENAME       NAME     FIELD-1   FIELD-2   FIELD-3   ICF CRD DBI   MODULE   QUAL        AUTH.-ID   73 TO 80    SEQ
-----------------------------------------------------------------------------------------------------------------------------------
(ASLRC   ) (RF) (SAMPLE  ) (S) (M4OLD   )                                                                                (ASLGEN'D)     2
-----------------------------------------------------------------------------------------------------------------------------------
  RUN     STMT  PARAMETER                                                                                                SEQUENCE   LIST
  NAME    TYPE    NAME                                  PARAMETER VALUE                                                  73 TO 80    SEQ
-----------------------------------------------------------------------------------------------------------------------------------
(ASLRC   ) (RP) (SYSDATE )        (061402                                                  )                            (ASLGEN'D)     3

** CLSAA00  TYPE 0  DEFINITION SAMPLE   IS BEING PROCESSED.                                                                  *****
-----------------------------------------------------------------------------------------------------------------------------------
  FILE    STMT    FILE    DELETE GLOSSARY RECORD RECORD RECORDS BUFFER MAPPED      MAPPING         MAPPED          SEQUENCE   LIST
  NAME    TYPE    IDENT    FILE   LEVEL   FORMAT  SIZE  PER BLK  SIZE   FILE    INIT      TERM     RECSIZ          73 TO 80    SEQ
-----------------------------------------------------------------------------------------------------------------------------------
(SAMPLE  ) (FD)             (R)             (F)    (  80)                                                                        4
-----------------------------------------------------------------------------------------------------------------------------------
  FILE    STMT  SEGMENT  DEL SGMT VRT              SUP SEG  KEY                AUTH          EXTERNAL             SEQUENCE   LIST
  NAME    TYPE   NAME    ETE NMBR SEG              DPL KEY CHAR                 ID          TABLE NAME            73 TO 80    SEQ
-----------------------------------------------------------------------------------------------------------------------------------
(SAMPLE  ) (LS) (ROOT    )     ( 1)                                                                                            5
-----------------------------------------------------------------------------------------------------------------------------------
  FILE    STMT   FIELD   DEL SGMT LEV FIELD  FIELD FLD SEG FLD DEC CNT OCRS OUTP EDIT      DATA BASE NAME/          SEQUENCE   LIST
  NAME    TYPE   NAME    ETE NMBR NUM LOCATN LNGTH TYP KEY RND PLC SEG NTIM EDIT LNGH      COLUMN HEADING           73 TO 80    SEQ
-----------------------------------------------------------------------------------------------------------------------------------
(SAMPLE  ) (L0) (NEWKEY1 )             (   1) (  4) (F) (1)                                                                      6
(SAMPLE  ) (L0) (NEWFLDA )             (   5) ( 16) (C)                                                                          7
(SAMPLE  ) (L0) (NEWFLDB )             (  21) ( 32) (C)                                                                          8
(SAMPLE  ) (L0) (DUEDATE )             (  53) (  6) (C)                                                                          9
(SAMPLE  ) (L0) (PRINCIPL)             (  59) (  8) (P)          (2)                                                            10

                                        DETAILED GLOSSARY BY NAME FOR                                          GLOSSARY PAGE     1
                                          FILE DEFINITION - SAMPLE

DATE CATALOGED -                                                                  DATE LAST UPDATED -
```

Figure D-112 Cataloging Example (Page 1 of 3)

```
CATALOGED BY -                                                                    LAST UPDATED BY -
EXPIRATION DATE -                                                                 DATE LAST USED -

FILE IDENTIFICATION =                                                             RECORD FORMAT = FIXED UNBLOCKED
NUMBER OF SEGMENTS IN FILE = 1                                                    RECORD SIZE = 80
NUMBER OF FIELDS IN FILE = 5                                                      BLOCK SIZE = 80


                                        ********************************
                                        *    SEGMENT   1, LEVEL 1     *
                                        *   SEGMENT NAME = ROOT       *
                                        ********************************

SEGMENT OCCURS N TIMES = 1                                        KEY FIELD 1 = NEWKEY1    TYPE = F   LENGTH = 4
SEGMENT SIZE = 66
NUMBER OF FIELDS IN SEGMENT = 5

-------------------------------------------------------------------------------------------------------------------------------
FIELD   FIELD   FIELD    FIELD   FIELD    DEC    CNT FIELD  EDIT CODES     EDIT    OUTPUT   LINE  DATA BASE NAME/
NAME    TYPE   LOCATION  LENGTH  RNDING  PLACES  FOR SGMT   ( ) ( ) ( )   LENGTH   WIDTH     NO   *** COLUMN HEADING ***
-------------------------------------------------------------------------------------------------------------------------------

DUEDATE   C        53       6                                                 6       6
NEWFLDA   C         5      16                                                16      16
NEWFLDB   C        21      32                                                32      32
NEWKEY1   F         1       4                                                14      14
PRINCIPL  P        59       8            2                                   21      21
** CLSGB00  TYPE 0  END OF FILE DEFINITION GLOSSARY.                                                                      *****

DEC 07, 2001  12.30.37                                                                                          PAGE      5


=================================================================================================================================
=                                                                                                                               =
=                                                  F I L E   S U M M A R Y                                                      =
=                                                                                                                               =
= JOBNAME: HANME02C                          STEPNAME: F446                                                                     =
=                                                                                                                               =
=  FILE NAME: M4OLD                          DSNAME: HANME02.HANME02C.JOB10682.D0000101.?                                       =
=  FILE NAME: M4LIB                          DSNAME: BUILDER.R140.M4LIB.R06#1                                                   =
=                                                                                                                               =
=================================================================================================================================

                                                  ******************************
                                                  *  PROC NAME - MAIN          *
                                                  *  INPUT STREAM PROCEDURE    *
                                                  ******************************
-------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                                 MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB       SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID           ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM       73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(ER)                                                                                (N)                              (ASLGEN'D)    11

    ;;BEGASL
    ;

    MAIN: PROC
       ;each master-file-record control procedure
    COUNTER: FIELD F 4 INIT 0
       ;
      IF N.NEWFLDA EQ 'ACME INC. '
         LET N.NEWFLDB = 'REPORTED BY SAMPLE PROCEDURE'
         LET T.COUNTER = T.COUNTER + 1
           IF T.COUNTER GE 5
              LET PF(F.EOF 1 1) = 'E'
           ELSE
              CALL REPORT SAMPREP1
           END
      END
    END PROC
    ;
-------------------------------------------------------------------------------------------------------------------------------
STMT   FIELD    FIELD FLD DEC OUTPT EDIT      INITIAL          LINE 1 OF        LINE 2 OF                          SEQUENCE   LIST
TYPE   NAME     LNGTH TYP PLC EDIT  LGTH       VALUE         COLUMN HEADING   COLUMN HEADING                       73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(TF)  (COUNTER )  (  4)  (F)                (0               )                                                     (ASLGEN'D)    12
-------------------------------------------------------------------------------------------------------------------------------
STMT SEQ  LOG CON .....OPERAND-A.....  OPER  ............OPERAND-B.............   ......RESULT......  PARTIAL FIELD  SEQUENCE   LIST
TYPE NO.  LEV CTR QLF FIELD   SEG-LVL  ATN   QLF FIELD   SEG-LVL  (OR CONSTANT)   QLF FIELD  SEG-LVL  STRT NUMB OPR  73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(PR)(100)          (N,NEWFLDA ) 001-1  (EQ)   (C,ACME INC.                    )                                    (ASLGEN'D)    13
(PR)(101)                              (NS)     (END                           )                                    (ASLGEN'D)    14
(PR)(102)                              (R )   (C,REPORTED BY SAMPLE PROCEDURE  ) (N,NEWFLDB ) 001-1                 (ASLGEN'D)    15
(PR)(103)          (T,COUNTER )        (+ )   (D,1                             ) (T,COUNTER )                       (ASLGEN'D)    16
```

Figure D-112 Cataloging Example (Page 2 of 3)

```
(PR) (104)          (T,COUNTER )        (GE)    (D,5                          )                                  (ASLGEN'D)    17
(PR) (105)                              (NS)      (108                        )                                  (ASLGEN'D)    18
(PR) (106)                              (R )    (C,E                          ) (F,EOF     )          ( 1) ( 1) (C) (ASLGEN'D)    19
(PR) (107)                              (GO)      (END                        )                                  (ASLGEN'D)    20
(PR) (108)                              (GO)      (SUB SAMPREP1               )                                  (ASLGEN'D)    21

                                                  ******************************
                                                  *  PROC NAME - SAMPREP1     *
                                                  *  INPUT STREAM PROCEDURE   *
                                                  ******************************
-------------------------------------------------------------------------------------------------------------------------------
STMT REPORT                                 MAX  SEL SUM VERT FORMS PAGE PAGE LINE REQ          BACK BR RNT AB      SEQUENCE   LIST
TYPE  DATE          REQUESTOR ID          ITEMS CTL RPT  SP  CNTRL WDTH HGHT NOS? TYP SET NAME CONTROL TMP FM      73 TO 80    SEQ
-------------------------------------------------------------------------------------------------------------------------------
(ER) (TODAY )                                                                      (S)                            (ASLGEN'D)    22

     SAMPREP1: REPORT TYPE SUBROUTINE,
                   COLUMNS  N.NEWFLDA  N.NEWFLDB  N.DUEDATE
                   FORMAT HEADINGS NAME
     END REPORT
-------------------------------------------------------------------------------------------------------------------------------
 ST SUM V  8  PG  PG   I SP  MAXIMUM TB COLUMN  DT  PG  STRT LIN SL RH .......SUBFILE........ C E A S R S N      C  SEQUENCE   LIST
 TY RPT S LPI WID HGT  M FRM LPP PGS TO HEADING POS POS PAGE NUM UB PN ENT    NAME   BLK/ FMT H F G T S U D      O  73 TO 80    SEQ
 MP     P             G            LT TYP POS         NUM      ML TD REC           BUF      C C S L T M S      N
-------------------------------------------------------------------------------------------------------------------------------
(E1)                                  (F)                                                                        (ASLGEN'D)    23
-------------------------------------------------------------------------------------------------------------------------------
          SP  Q                E  N  S  D  C  S  T  C  C  M  M  A  G  D Q PCT/RAT      P                PARTIAL   SEQUENCE   LIST
STMT SEQ  BF  L  FIELD   SEG-  N  P  E  E  T  U  O  U  N  A  I  V  M  E L  FIELD   RA- C      PICTURE    STRT NO   73 TO 80    SEQ
TYPE NO.  COL F  NAME      LVL  D  R  Q  S  L  B  T  M  T  X  N  G  R  C F  NAME    TIO T       EDIT      CHR CHR
-------------------------------------------------------------------------------------------------------------------------------
(R1)          (N,NEWFLDA ) 001-1                                                                                (ASLGEN'D)    24
(R1)          (N,NEWFLDB ) 001-1                                                                                (ASLGEN'D)    25
(R1)          (N,DUEDATE ) 001-1                                                                                (ASLGEN'D)    26

** MK4W204  TYPE 0  NUMBER OF MESSAGES PRINTED IS    2.                                                                  *****
                    M4OLD    -        6 RECORDS INPUT
                    M4REPO   -       16 RECORDS OUTPUT
                    M4INPUT  -       33 RECORDS INPUT
                    M4LIST   -      208 RECORDS OUTPUT

** MK4ED02  TYPE 0  REQUEST-SAMPREP1  REPORT-1  REQUESTOR ID-                                                           *****

06/14/02                                                                                                          PAGE 1

-------------------------------------------------------------------------------------------------------------------------------
      NEWFLDA                 NEWFLDB              DUEDATE
-------------------------------------------------------------------------------------------------------------------------------
  ACME INC.         REPORTED BY SAMPLE PROCEDURE      10202
  ACME INC.         REPORTED BY SAMPLE PROCEDURE      31702
  ACME INC.         REPORTED BY SAMPLE PROCEDURE      61702

** MK4FT03  TYPE 0  END OF REPORT.                                                                                        *****
```

Figure D-112 Cataloging Example (Page 3 of 3)

Chapter

# E Glossary

This glossary defines terms used throughout this guide. In cases where a term refers to fixed format statement input and VISION:Workbench for DOS input, the equivalent fixed format statement is indicated in parentheses. In cases where a term refers to fixed format entries and ASL codes, the equivalent fixed format entry is indicated in parentheses.

## A

**AA statement**
Allows you to enter comments with source statements.

**Absolute scale (graphics)**
All summary values are plotted relative to the total of the summary values so that if all bars are added, their sum would equal the maximum of the summary axis.

**Action code**
Identifies the action to be performed on the master file record field during transaction processing.❹ An entry on the Transaction Definition window (position [36] of a TL statement).

**AD statement**
Array definition statement. Specifies the array name and describes the record format.

**Address list, CALL statement**
Associated with each parameter list passed to VISION:Builder. There is an address list that is assembled and compiled as part of the program that issued the CALL statement. First in the list is the address of the first parameter; second is the address of the second parameter, and so on. The address list is composed of a set of fullword addresses. The last address in the list contains HEX 80 in its high order byte.

**Align files** Process by which key field values of records from different files are compared. This occurs in transaction processing❹ and file coordination.

**All records coordination** Processes all records in the coordinated file. It is also known as sequential coordination.

**Application run** Processes VISION:Builder requests, creates and maintains files, and generates reports.

**Arithmetic operator** Operator used in an arithmetic computation. The arithmetic operators available are addition (+), subtraction (-), multiplication (*), and division (/).

**Array** Structure that consists of data cells along two dimensions - rows (horizontal) and columns (vertical).

**Array record** Structured record that consists of data defined along two dimensions – rows (horizontal) and columns (vertical).

**ASL** Advanced Syntax Language that can be used in place of the fixed format PR statements for specifying procedural logic.

**Audit file** Optional output file that contains master file records deleted during a run.

**Auto grand summary** Specifies that a grand summary be printed for each field specified as a Summary entry on the Report Summaries window (En statement).

**Automatic function** A function that is performed by VISION:Builder without requiring any procedural coding.

**Automatic table lookup** A table lookup (see table lookup) is invoked by referencing, in a procedure or request, a field that is in the file definition as a table result field.

**Automatic revalidation** Process by which invalid fields in procedures or requests are automatically revalidated by VISION:Builder at the end of the procedure or request.

# B

**B0 statement** Defines the attributes of a field in an array. A B0 statement is required for each array field defined.

**Bn statement**

Defines the column heading text for a field defined by B0 statement. A field can have from one to nine lines of column headings, defined using B1-B9 statements. The Bn statement is optional.

**BA statement**

Specifies the table name, its type, and the input argument for automatic table lookup. BA statements are used in conjunction with B0 and Bn statements.

**Back branch**

Allows you to branch to the top of either a prior request of the same type or back to the top of the current request. Back branch is normally used with the Back Branch Control capability, specified on the Request Definition window (En statement), which limits the number of back branches and prevents infinite loops.

**Bar graph**

Called histograms, they incorporate bars of varying lengths to represent summarized data for a particular category. The bars can be plotted vertically or horizontally.

**Basic cycle**

The sequence of automatic operations and events that occur in every VISION:Builder application.

**Binary table**

Used in binary search. The table is stored on the common library in ascending order by argument. The binary search process consists of successively halving the table to find the argument that matches the value specified.

**Branch location**

Used with a branch operator. Specifies where the processing is being transferred.

In ASL, the branch locations for procedures are CONTINUE, NEXT MASTER, TYPE_1,❹ TYPE_2,❹ RETURN (or LEAVE), and a labeled procedure statement.

In fixed format syntax, the branch locations within a request are OUTPUT, END, and Sequence Number. The branch locations beyond a request are NEXT REQUEST, NEXT MASTER, REQUEST request-name, SUB request-name, SUB set-name, RETURN, TYPE 1, and TYPE 2.

**Branch operator**

Operator that transfers control from the current statement to another branch location.

In ASL, the branch operators are TRANSFER and GO [TO].

In fixed format syntax, the branch operators are GO, GS (select), and NS (non-select).

**Buffer size**

Amount of memory required for a block of records of undefined or variable length.

## C

**CALL statement**

Means by which control can pass from VISION:Builder procedures or requests to other systems or programs. In ASL procedures, means by which control can pass to reports, subroutine procedures, and subfiles. They are coded in a VISION:Builder procedure or request using a parameter string format. The return code and CSTATUS flags can be referenced to determine the status of a CALL statement.

**Category axis**

Bar graph axis that shows the categories of data being summarized. Categories represent either discrete values or ranges of values. In a vertical bar graph, the category axis is horizontal; in a horizontal bar graph, the category axis is vertical.

**Chained coordination**

Coordination in which there is a dependency relationship between coordinated files and only a single coordinated file in the chain is related directly to the master file.

**Checkpoint/restart**

Method to store, during a run, memory contents and input/output statements for restart procedures. The checkpoint options are specified in a CP statement of the run control group.

**Common library**

Data set used to store VISION:Builder definitions and requests (M4LIB).

**Control break** — Field used to indicate, by a change in its value, that a specified set of summary operations and/or print suppression is to occur. A control break field has a number from 1 to 9 in the Control Break entry on the Report Details window (Rn statement). Note that control break fields cannot be summarized.

**Column heading text** — Column heading to be printed when a field is output on a report.

**COMLIB** — See common library.

**Connector** — Combines relational expressions into a logical expression. The connectors available are AND and OR.

**Coordinated file** — File used as an auxiliary input file in a processing run. This file can be coordinated with the master file or another input file in an indexed, sequential, or user read manner.

**Coordinated request** — Request that contains a reference to a field in a sequentially coordinated file or to the ECORD flag.

**Coordinating field** — Used to directly access a record from a coordinated file. It can be any field in the application run, except the coordinated file being accessed. This field is compared to the key of the coordinated file record to perform coordination.

**Count field** — Field in a parent segment that counts the number of occurrences of its variable occurring dependent.

**CP statement** — Provides checkpoints during the execution of VISION:Builder runs. It is a component of the run control group.

**CR statement** — Specifies cataloged requests to be executed.

**CT statement** — Performs catalog/library maintenance functions.

**Cumulative (cum)** — Summary produced when a control break occurs, containing the total of all valid values of a specified field since the beginning of a report.

**Cycle** — Sequence of operations or events that occur for every instance of a record.

# D

**Data cell**
Element of an array record that is the intersection of a row and a column. A data cell contains one occurrence of one or more fields. All data cells comprise the same set of fields.

**DB statement**
Specifies the name of the run data group.

**Default**
Standard functions and formats provided by VISION:Builder when specifications are not made.

**Default create**
Automatically creates a master file record when a master record is not found during the transaction processing.❹ An entry in position [39] of a TD statement.

**Default edit length**
For fixed length fields, it is the field length, including edit characters, computed by VISION:Builder. For variable length fields, the length is the shorter of the report page width minus spaces before column or the defined field length.

**Definition/ maintenance run**
Catalogs or maintains definitions in the common library.

**Deleted record**
Master record that is read in from the old master file in an application run and is not written to the new master file, but optionally to the audit file.

**Deleted segment occurrence**
Segment occurrence that is read in from the old master file but is not written to the new master file.

**Delimiter**
System character. Indicates the end of a line of text, represents stored data characters in report picture editing, or is used as a separator in reporting specifications. It is specified in M4PARAMS and it is shown in this documentation as a pound sign (#).

**Delta increment (graphics)**
Specifies the size of one interval on the X or Y axis or of one range on the category axis. The difference between the start and end values are divided by the delta increment to determine the number of intervals.

**Dependent segment** Relative to a particular parent segment. A dependent segment is any segment that has another segment (its parent) directly above it in the hierarchy.

**Detail group** One of the components of a formatted report. The associated Fn statement indicates the number of lines of detail information that are to print on each page.

**Discrete category** Category axis with individual data points rather than a range of data.

**Discrete graph** Bar graph with individual data items plotted on the category axis.

**Displacement table** Used in a displacement (positional) search. The table arguments must be all integers. The table is searched against these integers. For example, given the number 9, the 9th table result value is retrieved.

**Duplicate keys** Key field values shared by two or more records in the same file or occurrences within the same segment type.

**Dynamic deletion** Technique of deleting a master file record by setting the DELETE flag❹ to 1. Also, a master file record segment can be deleted by setting the DELETE flag to 2 and emptying the key field of the occurrence to be deleted.

**Dynamic partial fielding** Uses the scan flags associated with the SL (scan left) and SR (scan right) operators to describe the matching part of the text and the parts to the left and right of the matching part.

# E

**En statement** Formats reports and specifies subfiles to be output.

**En statement (graphics)** Specifies the format of the graph, the graph data report and the statistics report. It also specifies the format of subfiles, if they are output.

**Edit length** Number of characters in a field, including editing of and output to a report.

**End line** Indicates that a field is to be the last one on a line of a report. Specified on the Report Details window (Rn statement).

**End of request** Branch location that specifies that the processing is being transferred to the end of a request (after the last Rn report section of a request).

**Entire record selection** Outputs an entire master file or coordinated file record to a subfile. Specified on the Subfile Definition window (position [43] on the En statement).

**ER statement** Allows you to describe the request (such as request name, report date, request type). It also gives you selection control option, back branch control, reinitialization of temporary fields, and parallel looping options. The default formatting capabilities can be overridden on the En statement.

**Execution time parameters** Overrides system defaults. Specified on the Application Parameters window (RP statement).

**External CALL statement** CALL statement from VISION:Builder to a user routine.

## F

**Fn statement** Controls the format of a report page.

**FD statement** File definition statement. Specifies the file name and describes the record format.

**Field** Smallest addressable unit of data. It has minimum length of one byte and has a unique name.

**Field conversion** Process that transforms a field value of one data type to a value of another data type (for example, character to packed).

**Field length** Number of bytes in a field.

**Field linkage** Allows concatenation of character fields. It uses the Cn operator, which links two specified fields, inserting n blanks (0-9) between them.

**Field location** Location relative to the beginning of a segment, of the first byte of a field.

**Field rounding**

Rounds numeric result fields of an arithmetic replacement, transaction update,❹ or a table lookup operation. If the field has n decimal places (n=0-9), the n+1 decimal place is half adjusted. That is, for a positive value result field, 5 is added to the n+1 decimal place, and the result is truncated; for a negative value result field, 5 is subtracted before truncation. Specified on the Segment Definition window (Y in position [32] on the L0 statement).

In ASL, rounding is specified using the LET command and the [WITH] ROUNDING operand.

In fixed format syntax, it is specified using +R, -R, *R, and /R.

**Field type**

Indicates the internal representation of data that can occupy the field. The types available are character (type C), zoned decimal (type Z), packed decimal (type P), fixed point binary (type F), floating point decimal (type E), variable length (type V), and time fields (type L, S, or P).

**File**

Collection of related records treated as a unit.

**File coordination**

Alignment between two input files.

**File definition**

Describes the logical structure of the file records to VISION:Builder. File definitions are cataloged in the common library in a definition/maintenance run.

**File name**

Uniquely identifies a file definition stored in the common library.

**Filling, output edit**

Values that print in every position of a field from the leftmost position until the first non-zero digit is encountered.

**Flags**

Fields that are used to communicate between VISION:Builder automatic processes and application procedures or requests. They are referenced with an F qualifier.

**Flat file**

Usually referred to as an unstructured file. A file with a root segment only.

**Floating output edit**

Values that print to the immediate left of the first significant digit of a field in a report.

**Forward branch** — A branch to a PR statement with a higher sequence number or to a request later in the chain, but not to the same request or one earlier in the chain. A forward branch requires no back branch control entries.

# G-H

**Generalized Data Base Interface** — This enables you to access and process data from virtually any Data Base Management System without having to be concerned about where the data is stored or in which format it is stored.

**Generalized System Interface** — Enables VISION:Builder to communicate with the database systems and user written routines or programs. Data can be passed to and from VISION:Builder, once it is defined to VISION:Builder. Control is passed from VISION:Builder to other systems by CALL statements with parameter lists.

**Generic key** — High order portion of a VISION:Builder record key containing characters that identify those records that are significant.

**Glossary** — Output of the definition/maintenance run. Lists the contents of a file definition, transaction definition, table definition, or array definition.

**Grand summary** — A summary of the specified field at the end of the report.

**Graph control field** — Similar to a level 1 control break field with page subtitles. Creates a separate data report, graph, and graph statistics report for each value of the graph control field.

**Graph report** — Lists each category with its individual summary detail field values for a bar graph or lists the contents of the two graph fields for a scatter graph. It also generates a total, cumulation, or count of the summary field values for each category. It can be suppressed with the suppress data report entry on the Jn statement.

**Graph modifier** — Identifies the axis that is defined by the Kn statement. Both the X and Y axis must be defined for a scatter diagram. C defines the category axis for a range bar graph, X defines the X axis for a scatter graph, and Y defines the Y axis for a scatter graph.

**Header group** — Component of a formatted report. Consists of title lines that contain one or more literals, field names, or flag names. The header group prints at the top of each page.

**Hierarchical coordination** — One coordinated file matches to the master, one or more coordinated files match to that first coordinated file, and so on.

**Hierarchical record** — See structured record.

# I-K

**Indexed coordination** — Records in the coordinated file are directly accessed (VSAM, IMS, DB2, and so on).

**Indexed file** — File that can be accessed directly. An index of keys determines the location of each record.

**Inherited segment occurrence** — Segment occurrence currently processed by a procedure or request is passed to a subroutine procedure or request.

**Input parameters, CALL statement** — Passes data to VISION:Builder from any routine invoking VISION:Builder. Defined in the linkage section, they are the data elements whose space is allocated by another program or the operating environment and whose location is passed to VISION:Builder by that program when it invokes VISION:Builder.

**Inserted segment occurrence** — Segment occurrence that is not read in from the old master file but is written to the new master file.

**Invalid field** — Field having illegal data that is not compatible with its data type, the result of division by zero, or a field not large enough to accommodate results. An invalid field appears on reports as an asterisk (if specified as such in M4PARAMS). A relational expression comparing an invalid field with itself is false.

**IT statement**
Item tracking statement. Provides the facility to maintain and report certain data about definitions (arrays, files, tables, transactions), requests, and logical run views. It belongs to the run control group.

**Jn statement (graphics)**
Defines the output of the request as one with graphics report output. It specifies the type of graph produced and the plot and axis characters used.

**Kn statement**
Defines the characteristics of an axis on a graph.

**Key field**
Field or fields used to uniquely identify a record or a segment.

**Key sensitivity**
Option coded in the PCB of an IMS database to secure it. When used, this option only returns the key field value to the PCB key feedback area.

**Keyword, CALL statement**
First entry in a CALL statement. All entries following the CALL keyword are preceded by at least one blank. (In fixed format, the CALL keyword is generally specified in positions [15-18].)

# L

**L0 statement**
Defines the attributes of a field. An L0 statement is required for each field defined in a record.

**Ln statement**
Relates the column heading text to a field. A field can have from 1-9 lines of column headings, defined using L1-L9 statements. The Ln statement is not mandatory; however, if used, the field must have a corresponding L0 statement.

**LA statement**
Identifies the type, input argument, and table value for automatic table lookup. LA statements are used with L0 and L1-L9 statements.

**LB statement**
Specifies an alias name in a run data group.

**Levels**
Horizontal layers of dependency. Up to nine levels of dependency are possible. They are numbered consecutively in an ascending order, top to bottom.

**Linear interpolation table lookup**

Table lookup procedure. Linearly estimates a result field value between two surrounding table results using the relationship of the table argument values to the two table argument values surrounding it.

**Linkage section**

Passes input parameters to VISION:Builder from any routine involving VISION:Builder. Fields defined in the linkage section are referenced by the qualifier V.

**LM statement**

Defines the relationship between a sequence related GDBI I/O command and the mapping request that processes that command.

**Logic level**

Number assigned to an AND/OR connector. Indicates the number of sets of parentheses around the connector when the logical expression is laid out.

**Logical data view**

Set of logical records combined together in an application; mechanism for combining data from *different* databases or access methods.

**Logical expression**

Consists of relational expressions connected by AND/OR operators. A logical expression with only AND operators is true only if all relational expressions connected by the AND operator are true. A logical expression with only OR operators is true only if at least one of the relational expressions connected by the OR operator is true.

**Logical record**

Defines all of the joins that are required by the application, regardless of database type or access method.

**Looping**

Automatic function. Processes multiple-occurring segment types or cells in an array. The first executed reference to any field in a lower level segment or array cell starts a loop. The range extends to the end of procedure if the loop started in a procedure (PR statement) or until the end of a report as defined in the Report Details window (as far as the Rn statements with the same value of n) if the loop began in a report.

**LR statement**

Specifies the logical relationship between two segments of a relational file definition.

**LS statement**

Names a segment in a record.

# M

**Mapped file definition** — Used in Generalized Database Interface (GDBI). Defines the logical structure of the record the application run will process and identifies the mapping requests that perform the I/O on the database.

**Mapping request** — Used in Generalized Database Interface (GDBI). Performs all I/O between VISION:Builder and the database.

**Master file** — Usually the primary file in an application run.

**Matched-record-only coordination** — Sequential coordination. Processes only those records in the coordinated file whose key match that of a record in the master file.

**Maximum** — Type of summary that contains the highest value a field has attained since the last control break at the level specified.

**Minimum** — Type of summary that contains the lowest value a field has attained since the last control break at the level specified.

**Missing or non-existent field** — Field that does not exist in the record or field whose value is not presently available. Missing fields print as a dash on a report or as a zero or blank in subfile records.

**MOSAIC** — Memory optimized (MOSAIC) processing brings, at most, one occurrence of each segment in storage at any one time (standard processing brings the entire logical record into the VISION:Builder work areas). In MOSAIC processing, a segment is not retrieved until it is referenced, either explicitly or implicitly.

**Multi-image report** — Report with more than one logical page printed horizontally on a physical page. This facility is specified on the Report Definition window (En statement) using the Width of Page and Number of Repeated Image entries.

**Multi-path nested loop** — Nested loop generated by referencing multiple segment types along two or more paths of a structured record.

**Multi-path parallel loop** Processes occurrences from two segments not in the same path. Each occurrence of one segment is processed only once with one occurrence of the other segment. The parallel looping entry in the ER statement activates this capability.

**Multiple file coordination** When there is no dependency relationship between any of the coordinated file records, the highest numbered coordinated file is read first.

# N-O

**New master file** File output as the master file by an application run. Its records are written from the new work area.

**No fit plot** Scatter graph that is plotted with no attempt to fit the X, Y pairs to a straight line or a curve.

**Non-print field** Controls the sequence or break of report records, but does not print on the report. Specified on the Report Details window (Y in position [26] of an Rn statement).

**Null field** A variable length (type V) field with a length of zero (no data).

**OC statement** Used when own-code hooks are activated and user written code has control of predetermined points during the run. It is a component of the run control group.

**Old master file** File input as the master file to an application run. Its records are read into the old work area.

**Operand** Identifies the field and qualifier referenced in a procedure for processing.

**Operator** Indicates the processing required on the operands referenced in a procedure.

**Output edit** Overrides the edit specifications in the file definition of the reported field. Specified on the Report Details window (Rn statement).

**Output, request of** Branch location that specifies that the processing is being transferred to the Rn (Report) sections of a request.

**Output width**
Equals the edit length (on the Segment Definition window or the L0 statement) or the column heading text size (on the Segment Definition window or the Ln statement), whichever is greater. It is automatically generated by VISION:Builder and prints in a file glossary.

**Own-code**
Facility that enables user written programs to be integrated with VISION:Builder at specific points in the processing cycle.

# P-Q

**Pn statement**
Used to enter preface information for a report or to initiate dynamic report line modification.

**PA statement**
Program analyzer statement. Provides numeric conversion messages, field name cross reference report, constant/pattern cross reference report, branch location cross reference report, file reference report, request execution trace report, and mapping request execution trace report. It belongs to the run control group.

**Page subtitle**
Prints at the top of each page. Any change in this field causes it to be printed as a subtitle on a new page after printing specified summary information related to the previous value of the field. Specified on the Report Details window (P in position [30] of the Rn statement), along with a control break specification.

**Parent segment**
Segment that has one or more segments directly below it in the hierarchy.

**Partial field**
Contiguous subset of characters in a field to be processed. Defined in procedures and reports.

**Path (leg)**
Set of segments that start with the root and contain one dependent segment at each level.

**Picture edit**
Consists of an EBCDIC characters, including embedded and leading blanks with stored field data characters represented by the system delimiter.

**Positional indicator** Required by some dynamic reporting keywords. Describes to VISION:Builder the format of the required data and where to find it on the output line. Positional indicators specify the entire field or its starting position and take the form POSnnn or nnn signifying three digits between 001 and 132.

**PR statement** Used for selection, calculation, and processing in a request.

**Preface page** Page preceding the body of the report. It is specified on the Preface Definition window (Pn statement).

**Preselection request** Available to process IMS databases. Generates a Boolean SSA to limit segment access to those required by the processing run. Specified on the Procedure Definition window (P in position [56] of the ER statement).

**Procedural table lookup** Table lookup that retrieves the table result. Specified by the LOOKUP command (TL operator) and an input argument in the procedure or request.

**Procedure** Means by which you select records from a file and specified data from records for computation or modification. A procedure accomplishes these tasks using ASL statements.

**Processing cycle** See basic cycle.

**Processing phase** Compiles VISION:Builder source statements, reads input files, outputs new files and produces a report file, if requested.

**Qualifier** Identifies type of field or file in which a field exists.

## R

**RA statement** Specifies that an array overdefines another array.

**Rn statement** Formats output and specifies the data on reports and/or subfiles.

**Rn statement (graphics)** Specifies the data fields for the graphics output on subfiles.

**Range category** Category axis with a range of data rather than discrete data points.

**RC statement** Identifies a particular run, specifies the input and output files and run-dependent parameters.

**Record** Set of data elements that are related to each other and treated as a unit.

**Record format** Identifies the physical format and access method used to read a file. The possible formats include: sequentially accessed fixed length records (F), sequentially accessed variable length records (V), sequentially accessed undefined records (U), indexed sequential records (I), variable ISAM records (J), key sequenced VSAM records (K), entry sequenced VSAM records (E), packed (P), and DL/I (D or H).

**Record size** Length of a logical record.

**Referential integrity** When data is changed in one table, it must be changed in all tables that reference it.

**Reject file** File of transaction records that are rejected during a transaction processing cycle.

**Relational expression** Consists of operand A, a relational operator, and operand B. A relational expression can be either true or false. A true relational expression results in the select condition, false in the non-select condition.

**Relational operator** Compares two fields. The relational operators available are EQ (equal), NE (not equal), LT (less than), GT (greater than), LE (less than or equal), GE (greater than or equal), CV (validate character field), DV (validate date field), SL (scan left), SR (scan right), and SN (scan not equal).

**Relative scale, graphics** Expands the scale and highlights differences among the summaries. All summary values are plotted relative to the longest or highest bar (that is, in proportion to the greatest summary value).

**Relative unit, graphics** Equals the ratio of the largest summary value to the width or height of the page.

**Replacement operator** Replaces the contents of a field with other content. The replacement operators available are R (replace), JL (justify left), JR (justify right), SS (scan), TL (table lookup), TN (table search, nearest), TB (binary table search, equal to or next larger), TS (binary table search, equal to or next smaller), TI (advanced table lookup), SS (scan and substitute), and Cn (combine).

**Report file** Contains data selected to be output on a report.

**Report phase** Immediately follows the sort phase or processing phase and produces a report image file suitable for printing.

**Request** Means by which you select records from a file, select specified data from records for computation or modification, and specify the output. A request accomplishes these tasks using fixed format statements with different functions. The statements in a request are tied together by a common name (the request name).

**Request chain** Sequence of requests of the same type that are processed in the order they are entered.

**Request type** Classification of a request, which states its timing in a processing cycle (for example, type 1 requests are processed before type 2 requests).

**Result (Operand C)** Only operand field of a procedure or request that can modify field values in the work areas. It is usually the result field of a calculation, text manipulation, or table lookup.

**RF statement** Describes files that are to be used in an application run in addition to the ones on the RC statement. The following files can be specified in the RF statement: coordinated files, subfiles, alternate report files, transaction files,❹ and arrays. The RF statement also links the file definition names with the files used.

**RG statement** Specifies the transaction groups❹ that are applied in a particular run. In the absence of an RG statement, all transaction groups❹ for the specified master file are used.

**Root segment** Single segment at the top of the hierarchy. It occurs only once per record.

**RP statement** Supplies miscellaneous run parameters (for example, execution time, storage optimization).

**RT statement** Specifies information for each subfile that is a relational table.

**Run control group** Determines which files are used in a run and sets some variable parameters for the run. The first set of statements in any VISION:Builder run.

**Run statistics** Message printed at the end of each VISION:Builder job step. Lists the number of records read from each input file and the number of records written to each output file. Run statistics also include the size of the common library and the space unused in it.

## S

**Scan only run** Application run that compiles VISION:Builder statements, performs the cataloged request maintenance, issues diagnostics, and produces reports with simulated data. It does not process the actual data. (An R entry in the scan/terminate control specification of a RC statement.)

**Scan operators** Operators include SL (scan left), SR (scan right) and SN (scan not equal) relational operators, and the SS (scan and substitute) replacement operator. They search a specified field (character or variable length) for a specified scan pattern. The SL and SR operators set several scan flags that are useful for dynamic partial fielding.

**Scan pattern** String of literal characters that is searched for by a scan operator. The location of the scan pattern within the scanned field sets scan flags.

**Scatter diagram** Relates two variables (X and Y) by plotting the points at which the variables intersect. Gives insight into the characteristics of the data. In addition, linear and second order fit characteristics of the data can be checked by plotting the fit on the graph.

**Segment** Entity of information (a set of fields) that are at various levels of the structure.

**Segment functions** Functions include FIND (FS), FIND FIRST (FF), FIND LAST (FL), and the RELEASE (RS) command. The FIND, FIND FIRST, and FIND LAST functions get a specific occurrence of a lower level segment without looping through the segment. The RELEASE command returns a segment occurrence referenced by the FIND, FIND FIRST, or FIND LAST function to an unreferenced status. A loop can be initiated on that segment type.

**Segment level** Number from 1 to 9. Describes the position of a segment in the structured record.

**Segment name** Value assigned to a segment in the Segment Definition window (positions [11-18] of an LS statement) of a file definition.

**Segment number** Number from 1 to 99. Uniquely identifies that segment (type) in the record hierarchy. In the hierarchy, segments are numbered in ascending order, top to bottom, left to right.

**Segment occurrence** Collection of field values (in a structured record) stored at the same level and having the same segment number. Segments at levels 2 through 9 typically occur multiple times in a record, either a fixed or variable number of times.

**Segment key** Field that uniquely identifies a segment within its parent.

**Segment type** Specifically numbered segment at a specific level in the hierarchical structure of the record. Used when referring to a segment within a definition, procedure, or request, not when referring to specific data records.

**SELECT statement** Statement that specifies the data query to be performed by DB2.

**Selection control** Facility that controls looping. Selects only the first occurrence of a segment that meets the criteria. Selection control is specified on the Request Definition window (ER statement).

**Sequence checking** Verifies that the master file records are sequenced by their record key field values. Specified on the Files Used window (RC statement).

**Sequential coordination** Coordination in which all involved files (including the master file) are sequentially accessible and sequenced by the same key. The coordinated files are matched against the master file.

**Sequential file coordination cycle** The sequence of operations and events that occur in a VISION:Builder application that includes sequential coordination.

**Sequential table** Table that is sequentially searched. The table arguments are compared to the input argument field in the order they are entered.

**Siblings** Two or more segments that have the same direct parent but are themselves independent.

**Significance loss** Message printed by graphics next to the label for the mean X and Y, standard deviation Y, standard error of estimate, and linear or second order fits when packed numbers with more than 15 integer digits are encountered. The packed numbers are converted to floating point values. Loss of significance is due to loss of accuracy when converting the floating number to character format for printing.

**Single path nested loop** Loop generated by referencing multiple segment types along a single path of a structured record. It can be in hierarchical order (down the path of the structure) or in an inverse order (up the path of the structure).

**Sort control field** Controls the sequence in which report records are printed. The field having the lowest sort sequence value is the major sort control field. Specified on the Report Details window (an Rn statement) as a number from 1 to 9 in the Sort Sequence entry.

**Sort phase** Usually immediately follows the processing phase. Sorts the report file for a subsequent reporting phase.

**Source statement listing** Formatted interpretation of the VISION:Builder requests and any associated errors.

**Standard subtitle** Control break field. Any change in its value results in printing the field on a line by itself after printing specified summary information related to the previous value. Specified on the Report Details window (an S in position [30] of the Rn statement).

**Start/end search** Limits the master file records available to the application run. Specified as key values on the Application Parameters window (RC statement).

**Statement** Line of VISION:Builder syntax that specifies a set of functions.

**Storage statistics** Message printed after the run statistics that lists the number of unused bytes of memory during the compiling and processing phases of VISION:Builder.

**Structured record** Record composed of more than one segment type.

**Subfile** Output data file that can be a subset of any input file (that is, the entire record is retrieved) or selected fields from any of the VISION:Builder work areas.

**Subroutine procedure or request** Procedure called by another procedure (code S in the request type of an ER statement). Subroutine procedures and requests are not invoked automatically by VISION:Builder; they can be invoked only by using the CALL statement (a branch operator with an appropriate branch location).

**Subroutine request set** Causes all individual subroutines within the set to be processed sequentially based on their order within the set. Composed of one or more subroutine requests, each having the same subroutine set name in an ER statement.

**Subtitle** Standard reporting feature. Prints a field on a line by itself (standard subtitle on page subtitle) when the value of the field changes. Specified on the Report Details window (position [30] of an Rn statement).

**Summaries** Collective name for all the arithmetic operations performed by VISION:Builder according to control breaks. The available summaries are total, cumulative, count, maximum, minimum, average, ratio, and percentage.

**Summary axis** Bar graph axis that summarizes the values of data for a category (such as the length of the bars). In a horizontal bar graph, the summary axis is horizontal; in a vertical bar graph, the summary axis is vertical.

**Summary label** — Normally prints to the left of print lines to denote summaries: total, cum (cumulative), count, max (maximum), min (minimum), avg (average), pct (percentage), and ratio.

**Summary level** — Number from 1 to 9. Relates the summary field to a control break field.

**Summary line** — Report line that lists one of the summaries specified in Report Details window (positions [31-36] of an Rn statement). It usually also includes at least one summary label.

**Summary only report** — Report that contains summary information only and no detail information. Only control break fields and fields that are summarized will print.

# T-U

**Tn statement** — Specifies the title printed on all numbered pages after the preface page.

**Tn statement (graphics)** — Specifies title lines for the graphics output.

**Table** — Collection of data that consists of a column of the table argument values and their respective result values. Tables are defined to VISION:Builder in the Table Definition window (TB and TE statements) and are cataloged in the common library.

**Table argument** — Field value of a table that is compared to the input argument to retrieve its associated table result.

**Table lookup** — Compares the input argument to the table argument to retrieve the table result. The table lookup can be performed procedurally by using the LOOKUP (TL) command in a procedure or automatically by referencing a table result field in a procedure or report, the result field having been predefined as such in the file definition.

**Table result** — Field value that is associated with its table argument. The table result is obtained from a table by comparing the input argument to the table argument.

**TB statement** Contains the attributes of the table argument and result fields.

**TD statement❹** Defines the following: transaction group, transaction identifier(s) in the group, master file definition that the transaction group uses, default record creation and segment insertions, and characteristics of the transaction file.

**TE statement** Defines the table argument and result entries.

**Temporary field** Field defined for the extent of a particular run. It is temporary in that it does not reside in any of the files in the application run. It is referenced with a T qualifier.

**TF statement** Defines a temporary field: the length, type, output editing, initial value, and up to two lines of column heading text. It is a statement type in a request.

**Time fields** Numeric field that consists of data belonging to a three unit system of measurement, such as hours, minutes, and seconds. The allowable data types for time fields are L, S, P, and U.

**TL statement❹** Defines actions to be applied to a master file with the transaction data.

**Top of request** Beginning of a request. Requests are always entered at the top, prior to executing any statements in the request.

**Total** Type of summary that contains the sum of all valid values of a field that have occurred since the last control break at the same level.

**Trailer group** Component of a formatted report. Consists of literals, field names, and summaries, intermixed in any order.

**Trailing, output edit** Values that print after the positive or negative values of a field on a report. Specified on the Segment Definition or Temporary Field Definition window (L0 or TF statement).

**Transaction cycle❹** The sequence of operations and events that occur in a VISION:Builder application that includes a transaction file.

**Transaction definition❹** Identifies and validates transaction records and applies them (if they are valid) according to the specified actions to the master file records.

**Transaction file❹** Updates the master file as specified by transaction definitions that identify its actions. The transaction file is always read sequentially. Any rejected transaction records can be written to the reject file.

**Transaction group❹** Set of transaction definitions applied against a particular master file. The transaction group can be specified on the Transaction Group Definition window (RG statement).

**Transaction processing❹** Applies transaction records (identified and validated by transaction definitions) against the master file records according to the actions specified in transaction definitions.

**Transaction record❹** Record that is action oriented to update a master file. The transaction record actions are defined through a transaction definition.

**Transaction identifier❹** Field in a transaction record that uniquely identifies it to perform specified update actions against a master file.

**Transaction request❹** Request that can reference the work areas before, during, and immediately after the master file record is updated. Transaction requests indicate the exit points in the transaction processing cycle. There are five transaction request types available: type 1, type 2, type M, type 3, and type 4.

**Transaction retry❹** Back branch capability used in transaction requests. Retry can only be used in type 2, 3, M, or 4 requests. Retry to type 1 is permitted from type 2, 3, M, or 4. Retry to type 2 is permitted from type 3 or M. Transaction retry requires use of the back branch control feature in the Request Definition window (En statement).

**UC statement** Initializes libraries, creates a backup copy of a library, reloads a library from a backup copy, condenses a library, copies items from one library to another, and copies all of the items in a library to another library. It belongs to the run control group.

**User-read coordination** Allows the user to control the reading of the coordinated file through procedural logic. The coordinated file can be read serially, directly, or both.

# V-W

**Variable length field** — Type of a character field that allows an increase or decrease in length of a field according to the amount of data it contains. It is also known as a type V field.

**Variable occurring** — Segment that has a variable number of occurrences. It requires a count field in its parent, unless it is in a IMS database.

**Virtual key** — File definition specification for a secondary index to a IMS database. It is used when the source is not the target segment or the source key is composed of non-contiguous fields.

**Virtual segments** — Working storage areas defined as part of a file definition of a database accessed through GDBI.

**WH statement** — Specifies additional SELECT statement, WHERE clause predicates, and other DB2 clauses that should be appended to the basic SELECT statement generated by VISION:Builder.

**Work areas** — Internal storage location that buffers input file records, output file records, tables, working storage, linkage areas, temporary fields, and VISION:Builder flags. Work areas are referenced with their respective qualifiers and field names.

**Working storage record** — Alternate form of temporary storage defined by one or more file definitions. A single level record resident in main storage.

# Index

## A


AA statement, E-1

DB2
  Relational database, 3-45, 3-47

Access methods, 6-3, 7-3

Action codes, E-1

Action codes for transactions, 4-6
  ADD TO, 10-16
  CLEAR, 10-17
  CONCATENATE, 10-17
  DELETE, 10-16
  INSERT, 10-16
  MATCH WITH, 10-16
  REPLACE, 10-17
  SUBTRACT FROM, 10-17
  UPDATE, 10-16
  XFER DATA, 10-17

AD statement, E-1

ADD TO, 10-16

Alias, DL/I, 3-30

Align files, E-2

All record processing, 6-26

All records coordination, E-2

Alternate indexing, 3-22

Alternate list file, 7-24

Alternate report file, 7-5

Analyzer, 20-1

Application components, 5-7

Application Cycle Overview, 5-3

Application cycle overview, 5-3

Application run, 2-20, 6-2, E-2
  Sample, 5-18

Arithmetic operations, 4-11, 4-12

Arithmetic operator, E-2

Array, E-2
  As a coordinated file, 13-27, 13-28
  Automatic looping, 13-13
  Branching, 13-11, 13-12
  Controlling looping, 13-9, 13-10
  Equivalence, 13-23, 13-24, 13-25, 13-26
  Loading in memory, 13-14, 13-15, 13-16, 13-17, 13-18, 13-19, 13-20, 13-21
  Looping, 13-5
  Nested looping, 13-6, 13-7
  Overdefining, 13-23, 13-24, 13-25, 13-26
  Parallel looping, 13-7, 13-8
  Processing, 13-13, 13-14, 13-15, 13-16, 13-17, 13-18, 13-19, 13-20, 13-21, 13-22, 13-23, 13-24, 13-25, 13-26, 13-27, 13-28
  Reporting from, 13-21, 13-22
  Selection control, 13-11
  Simple looping, 13-5, 13-6
  Subroutine processing, 13-12, 13-13
  To a subfile, 13-27

Array definitions, 2-20, 13-1, 13-2, 13-3, 13-4, 13-5

Array file, 2-15, 2-17
  Define, 2-20, 13-1, 13-2, 13-3, 13-4, 13-5
  Definition of, 2-15

Array Looping, 13-9

Array looping
Rules, 8-23
Array record, 2-20, E-2
Structure, 2-21
ASA carriage control, 9-22, 14-19, 14-21
ASL, 4-9, 4-17, E-2
Assembler, calling, 18-15, 18-16, 18-17
Asterisk, 4-6
Audit file, 2-17, 5-3, 5-4, 5-5, 5-6, 5-13, E-2
Definition of, 2-17
Audit record, 10-5
Authorization ID, 3-57
Auto grand summaries, E-2
Automatic
Conversions, 20-1, 20-2
Functions, 2-2, E-2
Grand summaries, 9-13
Revalidation, E-2
Table lookup, 2-20
Automatic Table Lookup, 12-6

## B

B0 statement, E-2
BA statement, E-3
Back branching, 11-13, 11-14, E-3
Bar graph, 14-27, 14-30, 14-31, 14-38, 14-39, 14-41, 14-42, 14-43, 14-44, E-3
Considerations, 14-43
Discrete, 14-31, 14-38, 14-39
Increase precision, 14-45
Range, 14-31, 14-38, 14-42, 14-43
Vertical axis, 14-48
Basic cycle, E-3
Basic processing
Common library, 5-2
Master file, 5-3, 5-5
Work areas, 5-5, 5-6
Binary table, 12-2, 12-3, E-3
Block size
OS/390 determined, 7-22
Bn statement, E-3
Boolean SSA, 16-9
Branch location, E-3
Branching, 11-10, 11-11
Back, 11-13, 11-14
Beyond a request, 11-10
Conditional, 11-5, 11-6, 11-7
Locations, 11-7, 11-8
Operators, 11-5, E-4
Procedure type, 15-6
Unconditional, 11-5
Using GS, 11-6
Using NS, 11-6
With subroutines, 11-11
Buffer size, 3-8, 3-12, E-4
IMS, 3-8
VSAM, 3-8

## C

CALL statement, E-4
Address list, E-1
Keyword, E-12
CALL statements, 18-1
Catalog, 5-2
Cataloging
Common library, 19-1
Category axis, E-4
Changing a file definition, 3-20
Character fields
Output edit length, 4-15
Checkpoint/restart, E-4
Child segment, 3-6, 4-19
CLEAR, 10-17
COBOL II, calling, 18-12, 18-13, 18-15
COBOL, calling, 18-15, 18-16
Collating key, 17-3

Collating reports, 17-1
Column heading character
  Suppress, 9-8
Column heading text, E-5
Column headings, 3-7, 3-19, 4-17, 9-4, 9-12
Column-heading section, 14-3
COLUMNS command, 9-3
COMLIB, E-5
Comma-Delimited File, 2-17, 7-25, 7-26
Comments, 3-11
COMMIT, 16-17
Common library, 2-2, 2-4, 2-10, 3-1, 3-20, 4-17, 5-2, 19-1, 19-2, 19-4, 19-5, 19-6, 19-7, E-4
  Definitions, 19-1, 19-2
  Item tracking, 19-6, 19-7
  Multiple, 19-7, 19-8
  Procedures, 19-5, 19-6
  Request groups, 19-5, 19-6
  Requests, 19-5, 19-6
  Run data groups, 19-4
Compare Operations, 4-11, 4-12
Compressed records, 3-91
Compression factor, 3-91
CONCATENATE, 10-17
Conditional branch, 11-5
Connectors, E-5
Constants, 4-10
contacting Computer Associates, web page, 1-6
Control break, 4-8, 9-9, 9-14, E-5
CONTROL command, 6-20, 9-19
Conversion, 4-11, 4-12
  *See also* Program analyzer, 20-1
Coordinated File
  All record processing, 6-37, 6-38, 6-39
  Chained coordination, 6-39, 6-40, 6-41, 6-43, 6-44
  Nested processing, 6-33, 6-34, 6-35
  Parallel processing, 6-31, 6-32, 6-35, 6-36, 6-37
Coordinated file, 2-15, 4-19, 6-2, 6-10, 6-11, 6-15, 6-19, 6-20, 6-21, 6-22, 6-23, 6-24, 6-26, 6-27, 6-28, 15-3, E-5
  All record, 6-24, 6-26
  Defining, 6-10
  Definition of, 2-15
  Directly, 6-53
  ECORD flag, 6-11
  From an array, 13-27, 13-28
  Generic keys, 6-55
  Nested processing, 6-29, 6-30
  Parallel processing, 6-29
  Processing, 6-24
  Qualifier, 6-10
  Sequential, 6-17, 6-18, 6-19, 6-20
  Serially, 6-52
  Update from subfile, 7-10
Coordinated request, E-5
Coordinating field, E-5
Coordination
  Chained, E-4
  Indexed, 6-11, 6-12, 6-13, 6-14
  Matched-record-only, E-14
  Multiple file, E-15
  Sequential, 6-29, 6-30, 6-31, 6-32, 6-33, 6-34, 6-35, 6-36, 6-37, 6-38, 6-39, 6-40, 6-41, 6-43, 6-44, 6-45, 6-46, E-22
  User, 6-46, 6-47, 6-48, 6-49, 6-50, 6-51, 6-52, 6-53, 6-54, 6-55, 6-56, 6-57, 6-58, 6-59
Count field, E-5
CP statement, E-5
CR statement, E-5
CREATE (C), 10-16
Cross reference report, 20-4, 20-5, 20-6, 20-7, 20-8, 20-9, 20-11, 20-12, 20-13, 20-14, 20-15, 20-16, 20-17, 20-22, 20-30, 20-31, 20-32, 20-33, 20-34, 20-36
  Branch location, 20-14, 20-15, 20-33, 20-34
  Constant/pattern, 20-12, 20-13, 20-32
  Field name, 20-8, 20-9, 20-11, 20-12, 20-22, 20-30, 20-31
  File name, 20-15, 20-16, 20-17, 20-23, 20-34, 20-36
CT statement, E-5
Cumulative (cum), E-5

Cycle, E-5

## D

Dash sign, 4-19
Data cell, 2-20, E-6
Data code, 3-92
Data control field, 3-92
Data definitions, 2-2, 3-1
Data structures, 2-3, 3-1, 8-1
Data synchronization, 3-70
Database access methods, 3-2
Database description
  DBD, 3-23
Database management systems, 6-3, 7-3
DB statement, E-6
DB2
  Column functions, 3-47, 3-84
  COMMIT, 16-17
  Considerations, 16-17
  Foreign keys, 16-16
  Joins, 3-45
  NULL columns, 16-18
  Relational support, 16-12, 16-13
  Reserved words, C-1
  Sample applications, A-1
  Scalar function, 3-47
  SELECT, 3-73
  Uniqueness of rows, 16-16
  Validation against catalog, 16-16
  WHERE clauses, 3-70
DB2 statements, 16-13, 16-14, 16-15, 16-16
  DELETE statement, 16-16
  INSERT statement, 16-15
  SELECT statement, 16-14, 16-15
  UPDATE statement, 16-15, 16-16
DBD, 3-23
  Processing options, 16-2
  Sample, 3-29
  Write from file definition, 3-25, 3-27, 3-28, 3-29
  Write to file definition, 3-29, 3-30, 3-31
Decimal, 3-17
DECIMALEDIT, 14-20
Default, E-6
Default create, E-6
Default edit length, E-6
Defaults, VISION:Builder, 9-4, 9-5
Define a field, 3-17
Define a logical record, 3-46
Define a segment, 3-13
Define a subfile, 7-6, 7-7
Define a table, 12-5, 12-6
Define a temporary field
  Explicitly, 4-17
Define a transaction, 10-7, 10-10, 10-13, 10-14, 10-15, 10-16
Define a VSAM file, 3-20
Define an array, 2-20, 13-1, 13-2, 13-3, 13-4, 13-5
Definition maintenance run, 5-2
Definition of
  Array, 13-1
  Array file, 2-15
  Array record, 2-21
  Audit file, 2-17
  Auto grand summaries, 9-13
  Bar graph, 14-30
  Bar graph axis, 14-31
  Common library, 5-2
  Conditional branch, 11-5
  Control break, 9-8
  Coordinated file, 2-15, 6-10
  Dependent, 3-5
  ECORD flag, 6-11
  Field, 3-5
  Field types, 4-1
  Fixed length, 3-5
  Fixed occurring, 3-6
  Flat file, 3-1
  Generic key, 6-51

Indexed coordination, 6-11
Key uniqueness, 6-4
Level number, 3-6
Levels, 9-9
Looping, 8-2
Master file, 2-15
Match file, 6-18
Match key, 6-18
Multi-path nested loops, 8-4
Multi-path parallel loops, 8-4
Page subtitles, 9-13
Parent/child, 3-6
Path or leg, 3-6
Rejected file, 2-17
Relative scale, 14-31
Repeating subtitles, 9-14
Scale, 14-31
Scatter diagram, 14-28
Segment, 3-6
Segment key, 3-6
Segment occurrence, 3-6
Segment type, 3-6
Simple file, 3-1
Single-path nested loops, 8-4
Subfile, 2-17
Subtitles, 9-13
Table, 12-1
Table lookup, 12-1
Transaction definition, 10-1
Transaction file, 2-15
Transaction procedures, 15-6
Transaction records, 10-1
Type A requests, 11-4
Type E requests, 11-4
Type N requests, 11-4
Type S requests, 11-4
Unconditional branch, 11-5
Variable length, 3-6
Variable occurring, 3-6
Working storage, 3-93

Definition/maintenance run, 2-20, 3-20, E-6
Sample, 3-93, 3-94

DELETE flag, 5-13, 10-16, 15-12
MOSAIC processing, 16-6
Transaction processing, 10-16

DELETE statement, 16-16

Deleted record, E-6

Deleted segment occurrence, E-6

Delimiter, E-6

Dependent segment, E-7

Descriptor word, 3-91

Detail group, E-7

DIRECT BY keyword, 6-12

Direct-Access
From subfile, 7-10

Direct-access, 6-5

Discrete category, E-7

Discrete graph, 14-31, E-7

Displacement table, 12-4, 12-5, E-7

DL/I
IMS, 3-8

Documentation
Overview, 1-1

Double-spacing on reports, 9-7

Duplicate keys, E-7
Eliminating, 15-41

Duplicates
Suppress, 3-70, 3-71

Dynamic deletion, E-7

Dynamic partial fielding, E-7

Dynamic report line modification, 14-19, 14-20, 14-21, 14-22, 14-23, 14-24, 14-25, 14-26
Examples, 14-24, 14-25
Multi-image, 14-25, 14-26
Processing, 14-20
Restrictions, 14-24
Storage considerations, 14-23
Syntax, 14-23

Dynamically delete
MOSAIC processing, 16-6

Dynamically deleted segments
Subfile, 7-13

## E

ECORD flag, 6-11, 6-19
Indexed coordination, 6-12
Partial fielding rule, 6-11
Sequential coordination, 6-19
Transaction processing, 15-15
Edit length, E-7
En statement, E-7
En statement (graphics), E-7
END branch, 11-7
End line, E-8
End of request, E-8
End search, 6-6
Entering a file definition, 3-10
Entire record selection, 7-9, E-8
EOF flag, 5-6, 5-9
ER statement, E-8
ESDS, 3-21
Execution time parameters, E-8
Execution trace report, 20-4, 20-5, 20-6, 20-7, 20-8, 20-17, 20-18, 20-19, 20-20, 20-21, 20-24, 20-36, 20-37
Mapping request, 20-20, 20-21, 20-36, 20-37
Request, 20-17, 20-18, 20-19, 20-24
Explicit temporary fields, 4-17, 4-18
Extended packed decimal fields, 4-4
Extended report
Dynamic report line modification, 14-19
External CALL, E-8
External calls
Assembler, 18-15, 18-16, 18-17
Call considerations, 18-12
Call status, 18-11, 18-12
COBOL, 18-15, 18-16
COBOL II, 18-12, 18-15
Execution, 18-21, 18-22
FORTRAN, 18-17
Implementation, 18-20
IMS, 18-22, 18-23
PL/I, 18-17, 18-18, 18-19
Return codes, 18-11
Specifying, 18-8, 18-9, 18-10, 18-11
External program linkage, 18-1, 18-2, 18-3, 18-5, 18-6, 18-8, 18-9, 18-10, 18-11, 18-12, 18-14, 18-15, 18-16, 18-18, 18-20, 18-22, 18-23

## F

FD statement, E-8
field output edit, 9-16
Fields, E-8
Conversion, 4-11, 4-12, 4-13, E-8
Define, 3-17
Fixed point, 4-3
Floating point, 4-3
Length, E-8
Linkage, E-8
Location, E-8
Missing, 4-19
Overdefine, 3-8
Overflow, 4-18, 4-19
Packed decimal, 4-3
Qualifiers, 5-5
Rounding, E-9
Sensitivity, 3-24
Starting position, 14-21
Time, 4-3
Type E, 4-3
Type F, 4-3
Type L, 4-3
Type P, 4-3
Type S, 4-3
Type U, 4-3
Type Z, 4-3
Types, 4-1, 4-8, E-9
Variable length, 14-18
Zoned decimal, 4-3
FILE AUDIT command, 7-22

FILE command, 16-15
File coordination, E-9
File definition, 2-3, 2-8, 3-6, 3-9, 3-20, E-9
  Change, 3-20
  Enter, 3-10
  Flat, 3-1
  instream, 19-2
  Mapped, E-14
  Master file, 6-5
  Override, temporary, 3-20
  Record compression, 3-90
  Report design, 9-4
  Simple, 3-1
  VSAM, 3-20
  Write from DBD, 3-29, 3-30, 3-31
  Write from PCB, 3-29, 3-30, 3-31
  Write to DBD, 3-25, 3-27, 3-28, 3-29
  Write to PCB, 3-25, 3-27, 3-28, 3-29
File glossary, 3-8
File input
  Coordinated files, 6-10
  Master file, 6-4
  Transaction file, 6-9
File maintenance, 2-2
File name, E-9
File organization, 3-2
File output
  alternate report output methods, 7-25
    comma-delimited data, 7-26
    HTML document, 7-27
    plain text file, 7-26
    raw data output, 7-27
    tab-delimited data, 7-26
  Audit, 7-22
  M4REPO, 7-4, 7-5
  Master file, 7-3, 7-4
  Rejected, 7-22
  Report files, 7-4, 7-5
FILE REJECT command, 7-22
FIND function, 8-21, 8-22, 8-23
Fixed length record, 3-5
Fixed length segment, 3-5
Fixed occurring, 3-6
Fixed occurring segment, 3-8
Fixed point fields, 4-3
  Output edit length, 4-15
Flag fields, 5-6, E-9
  DELETE, 5-6
  ECORD, 6-19
  EOF, 5-6
  Formatted sectional reporting, 14-4, 14-14
  LSTATUS, 8-24
  MISSPASS, 6-19
  MSTATUS, 20-21
  STRAN, 15-12
  XTRAN, 15-12
Flat file, 2-1, E-9
Floating output edit, E-9
Floating point fields, 4-3
  Output edit length, 4-16
Fn statement, E-8
Follow-up pass, 6-22, 6-26
FORMAT command, 9-3, 9-15
FORMAT command keywords, 9-4
Formatted reporting, 2-21, 14-13, 14-16, 14-17, 14-18, 14-19
  Variable length fields, 14-18
  When to use, 14-19
Formatted sectional reporting, 2-21, 14-1, 14-2, 14-3, 14-4, 14-5, 14-6, 14-7, 14-8, 14-9, 14-10, 14-11, 14-19
  Column-heading, 14-3
  Flag fields, 14-4, 14-14
  Page-title, 14-3
  Summary, 14-3
  Syntax, 14-2
  Variable length fields, 14-18
  When to use, 14-19
FORTRAN, calling, 18-17
Forward branch, E-10
FREEMAIN SIZE, 16-11

FS operator
FIND function, 8-21

## G

GDBI, 6-3, 7-3, E-10
Generalized System Interface, E-10
Generic key, 6-51, E-10
Glossary, 3-8, 3-9, E-10
GO branch, 11-8
Grand summary, E-10
GRANDSUMS keyword, 9-13
Graph control field, E-10
Graph modifier, E-11
Graph report, E-10
Graphics, 14-27, 14-29, 14-30, 14-31, 14-32, 14-34, 14-35, 14-37, 14-38, 14-39, 14-41, 14-42, 14-43, 14-44, 14-46, 14-47, 14-48
Absolute scale, E-1
Bar graph, 14-30, 14-31, 14-38, 14-39, 14-41, 14-42, 14-43, 14-44
Delta increment, E-6
Label lines, 14-49
Page layout, 14-47, 14-48
Relative scale, E-18
Relative unit, E-18
Scatter diagram, 14-27, 14-29, 14-32, 14-34, 14-35, 14-37, 14-46
Technical considerations and restrictions, 14-44
Vertical axis, 14-48
GS branch, 11-8
GSI, 6-3
GSI call, 18-1

## H

Header group, E-11
Headings
Column, 9-4, 9-12
Hierarchical
Coordination, E-11
Important points, 3-5
Record, E-11
Record structure, 3-1
Structure, 3-5, 8-1
HTML Document, 2-17, 7-25, 7-27

## I

ICF entry, 6-12
IGCPAL2, 20-29, 20-30, 20-31
IGCPAL3, 20-32
IGCPAL4, 20-33, 20-34
IGCPAL5, 20-34, 20-36
IGCPAL6, 20-36, 20-37
IGCPALFT, 20-27, 20-28, 20-29
IGCPALRS, 20-27, 20-28, 20-29
Implicit temporary fields, 4-6, 4-17
IMS
Alias, 3-30
And VISION:Builder, 3-23
Batch message processing, 3-25
Buffer size, 3-8
Considerations, 16-11
Database description (DBD), 3-23
DBD name, 3-11
DBD/PCB creation from file definition, 3-25
File definition, 3-25
File definition from DBD/PCB, 3-29
File ID, 3-11
From subfile, 7-10
Glossary headings, 3-26
Key fields, 3-23
KEYLEN (of the PCB), 3-28
LANG (of the PCB), 3-28
LCHILD (of the DBD), 3-29
NAME (of the DBD), 3-27
PARENT (of the DBD), 3-29
Preselection, 3-24
Processing, 3-23, 3-24, 3-25, 16-1, 16-2
PROCOPT (of the PCB), 3-27

Program communication block (PCB), 3-23
Program specification block (PSB), 3-23
PSBNAME (of the PSB), 3-28
Search fields, 3-24
Segment keys, 3-24
Segment ordering, 3-25
SENSEG (of the PCB), 3-28
Standard processing, 16-3
Updating considerations, 15-42
Variable length segments, 3-23
IMS MOSAIC processing, 3-24
IMS secondary indexing
Access, 3-31
DBD, 3-33
Inverted structure, 3-32
PCB, 3-32
Preselection, 3-33
Processing, 3-33
Restrictions, 3-32
XDFLD SRCH field, 3-32
IMS standard processing
HDAM, 16-3
IMS, calling, 18-22, 18-23
Index
Logical data, 2-8
Indexed coordination, 6-11, 6-12, 6-13, 6-14, 6-15, E-11
Indexed file, E-11
Indexing
Secondary, 16-10
Inherited segment occurrence, E-11
Input files, 5-13
Access methods, 6-3, 7-3
Arrays, 6-5
Automatic functions, 6-2
Database management systems, 6-3, 7-3
Processing flow, 6-4
Update, 5-8
Input parameters, CALL statement, E-11
INSERT, 10-16
Insert
Blanks, 14-19
Zeros, 14-19
INSERT statement, 16-15
Inserted segment occurrence, E-11
Instream File Definitions, 19-2
Invalid fields, 4-6, 4-18, 4-19, E-11
IT statement, E-12
Item tracking, 19-6, 19-7


## J


JCL
Report manager, 17-12
Jn statement (graphics), E-12


## K


Key field, 4-11, 6-22, 6-25, 10-5, E-12
Key sensitivity, 16-11, E-12
Key sequence, 3-7, 3-15
Key uniqueness, 6-4
Kn statement, E-12
KSDS, 3-21


## L


L0 statement, E-12
LA statement, E-12
Labels, 4-14, 9-12, 9-13, 9-16
LB statement, E-12
Leg, 3-3, 3-6
LET, 4-17
LET statement, 4-17
Levels, E-12
LINE SIZE, 14-21
Line spacing, 9-15
Linkage section, 18-1, 18-2, 18-3, 18-5, 18-6, 18-8,

18-9, 18-10, 18-11, 18-12, 18-14, 18-15, 18-16, 18-18, 18-20, 18-22, 18-23, E-13
Defining, 18-5, 18-6, 18-7
Parameter list, 18-2, 18-3, 18-4
*See also*External calls, 18-1
VSE considerations, 18-4
LM statement, E-13
Ln statement, E-12
LOCATE, 13-9, 13-10
Logic level, E-13
Logical data, 2-8
Example, 3-57, 3-58, 3-62, 3-65, 3-66, 3-68, 3-70, 3-71, 3-73, 3-75, 3-77, 3-78, 3-79, 3-82, 3-83, 3-84, 3-85
Logical data view, E-13
Logical expressions, E-13
Logical file
Unsequenced, 3-68
Logical record, E-13
Define, 3-46
Mapping, 3-47
logical record
Overview, 3-43
Logical status flag (LSTATUS), 8-24
Logical view, 3-1
LOOKUP, 2-20, 12-8
Looping, 3-1, 3-5, 8-2, 8-3, 8-4, 8-21, E-13
Examples, 8-12, 8-13, 8-14, 8-15, 8-16, 8-17, 8-18, 8-19
Implicit on parent segment, 8-23
In a report, 8-15
Inverse order, 8-16, 8-17
Multi-path nested, 8-8, 8-9, 8-10, 8-17, 8-18, 8-19, E-14
Multi-path parallel, 8-11, 8-12, 8-17, 8-18, 8-19, E-15
Range of a loop, 8-14
Segment operators, 8-21
Selection control, 8-20
Simple, 8-4
Single path nested, E-22
Single-path nested, 8-5, 8-6, 8-7, 8-12, 8-13, 8-16
Subroutine, 8-24, 8-25
looping
multi-path parallel looping
multi-path nested looping
examples, 8-20
LR statement, E-13
LS statement, E-13
LSTATUS flag, 8-24

## M

M4AUDIT, 5-3
M4CORDn, 6-10
M4INPUT, 6-59
M4LIB
Common library, 2-10
M4LIST, 7-23
M4LIST1, 7-24
M4NEW, 5-3
M4OLD, 5-3
M4PAOUT, 20-4, 20-5, 20-6, 20-7, 20-8
Generate, 20-4, 20-5, 20-6
Processing, 20-6, 20-7, 20-8
M4PARAMS, 7-4
M4REPn, 5-3
M4REPO, 5-3
M4SUBFn, 7-7
M4WORKnn, 3-93
Mailing labels, 14-25
Mapping data, 2-8, 3-2
Mapping request, E-14
Master file, 2-2, 3-5, 4-8, 6-2, 6-4, 7-3, 7-4, 7-13, 7-22, 10-4, 10-5, 10-6, 10-7, 10-10, E-14
Definition of, 2-15, 6-4
Deleted records, 7-22
Direct-access, 6-5
Dynamically deleted segments, 7-13
Eliminate duplicate keys, 15-41

End search, 6-6
Header/trailer processing, 15-41
MOSAIC processing, 6-8
One buffer, 6-7
Pre-summarization, 15-40
Preselection, 6-8
Processing, 5-3, 5-4
Processing flow, 5-11, 5-13
Processing options, 6-5, 6-6, 6-7, 6-8
Sequence checking, 6-5
Sequential access, 6-5
Start search, 6-6
Update-in-place, 6-6, 10-6
Uses other than updating, 15-39
WH statement, 6-8
Work areas, 5-5
MATCH (M), 10-16
Match file, 6-19
Match key, 6-19, 6-22, 6-26
Match only, 6-22, 6-23, 6-26, 6-28
Maximum, E-14
Memory optimized processing, 16-5
*See also*MOSAIC processing, 16-5
Minimum, E-14
Missing fields, 4-19
Missing or non-existent field, E-14
MISSPASS flag, 6-19
MK4W212, 16-6
MK4W225, 16-6
MK4W226, 16-6
More Attr, 3-17
MOSAIC, E-14
MOSAIC processing, 6-8, 16-5, 16-6, 16-7, 16-8, 16-9, 16-10
Dynamically delete, 16-6
Example, 16-8, 16-9
IMS, 3-24
Multiple positioning, 16-5
Single leg, 16-5
Storage utilization, 16-11
Technical considerations, 16-11
Updating, 16-6
MOVEPAGENO, 14-21
MS Processing, 16-1
MSTATUS flag, 20-21
Multi-column keyed segment, 3-86, 3-87, 3-88, 3-89, 3-90, 16-18, 16-19, 16-20
Multi-path nested loops, 8-8, 8-9, 8-10
Multi-path parallel loops, 8-11, 8-12
Multiple file, 6-38
Multiple images, 9-22
*See also*Dynamic report line modification, 14-19

## N

Nested loop
Definition of, 8-4
Inverse order, 8-6
New master file, 2-17, E-15
NEXT MASTER branch, 11-8
NEXT REQUEST branch, 11-8
No fit plot, E-15
NOCALC, 14-20
Non-print field, E-15
Non-repeating data, 3-92
Normalized tables, 3-63
NOSPACELINES, 14-20
NS branch, 11-8
NULL columns, 16-18
Null fields, E-15
Numeric conversion, 20-2
Messages, 20-2

## O

OC statement, E-15
Old Master File, 2-15

Old master file, E-15
One buffer, 6-7
Operand, E-15
Operand C, E-19
Operating environments, 1-6
Operator, E-15
Optimization
  Program analyzer, 20-1
Outer joins, 3-45
Output
  Request of, E-15
OUTPUT branch, 11-7
Output edit, 3-7, 4-14, 4-15, 4-16, 4-17, 9-4, 9-16, E-15
  Filling, E-9
  Length, 4-14
  Trailing, E-25
Output files, 2-16, 2-17, 5-13
Output width, E-16
Overdefine, 3-48
Own-code, E-16

## P

PA statement, E-16
Packed decimal fields, 4-3
Packed fields
  Output edit length, 4-15
Page characteristics
  Logical page, 9-20
Page layout, 9-21, 9-22
Page number, 14-21
Page subtitle, E-16
Page width, 14-21
Page-title section, 14-3
PAL
  Program analyzer, 20-1
Parent segment, 3-3, 3-5, 3-6, 4-19, E-16
Partial field entries, 4-20, 5-9
Partial fields, E-16
Path, 3-3
Path (leg), E-16
PCB, 3-23
  MOSAIC processing, 16-5
  Processing options, 16-2, 16-6
  Secondary indexing, 3-32
  Source listing, 3-28
  Write from file definition, 3-25, 3-27, 3-28, 3-29
  Write to file definition, 3-29, 3-30, 3-31
PERCENT, 14-20
Performance
  Program analyzer, 20-1
Physical files, 3-2
Picture edit, E-16
PL/I, calling, 18-17, 18-18, 18-19
Plain Text File, 2-17, 7-25, 7-26
Plus sign, 4-14, 4-19
Pn statement, E-16
Positional indicator, E-17
PR statement, E-17
Preface page, E-17
Preprinted forms, 14-17, 14-18
Preselection, 6-8, 16-9, 16-10
  Multiple SEGMENT / SSA pairs, 16-9
  Request, E-17
  Search fields, 3-24
  Secondary indexing, 3-33
Primary file
  Master file, 6-4
Procedures, 3-5, 5-8, 10-1, E-17
  After update, entire record view, 15-7
  After update, transaction view, 15-7
  Before master file read, 15-7
  Cataloging, 19-5, 19-6
  Each Master Record, 5-8
  End of File, 5-8
  Initialization, 15-6

Main, 5-8
Reports, 5-9
Subroutine, 5-8, E-23
Transaction error handling, 15-7
Transaction file and master file aligned, 15-6
Transaction record initial validation, 15-6
Processing, 2-2, 2-8, 4-10, 10-1, 10-16
Array, 13-13, 13-14, 13-15, 13-16, 13-17, 13-18, 13-19, 13-20, 13-21, 13-22, 13-23, 13-24, 13-25, 13-26, 13-27, 13-28
Components, 5-1
Data structure, 8-1
Flow for an application, 2-18
Hierarchical structure, 8-1
IMS, 3-23, 3-24, 3-25, 16-1, 16-2
Master file, 5-3, 5-4
Master file options, 6-5, 6-6, 6-7, 6-8
Memory optimized, 16-5
*See also*Program analyzer, 20-1
Storage considerations, 16-11
Technical considerations, 16-11
Work areas, 5-4, 5-5
Processing flow
Input files, 6-4
Processing phase, E-17
PROCOPT, 16-2
Program analyzer, 20-1, 20-2, 20-4, 20-5, 20-6, 20-8, 20-9, 20-11, 20-12, 20-13, 20-14, 20-15, 20-16, 20-17, 20-18, 20-19, 20-20, 20-21, 20-22, 20-23, 20-24, 20-26, 20-27, 20-28, 20-29, 20-30, 20-31, 20-32, 20-33, 20-34, 20-36, 20-37
Automatic conversions, 20-1
Cross reference, 20-4, 20-5, 20-6, 20-7, 20-8, 20-9, 20-11, 20-12, 20-13, 20-14, 20-15, 20-16, 20-17, 20-22, 20-30, 20-31, 20-32, 20-33, 20-34, 20-36
Execution trace, 20-4, 20-5, 20-6, 20-7, 20-8, 20-17, 20-18, 20-19, 20-20, 20-21, 20-23, 20-24, 20-36, 20-37
IGCPAL2, 20-29, 20-30, 20-31
IGCPAL3, 20-32
IGCPAL4, 20-33, 20-34
IGCPAL5, 20-34, 20-36
IGCPAL6, 20-36, 20-37
IGCPALFT, 20-27, 20-28, 20-29
IGCPALRS, 20-27, 20-28, 20-29
Program maintenance, 20-21, 20-22, 20-23, 20-24
Report run, 20-24, 20-27, 20-28, 20-29
Report summary files, 7-16
*See also*M4PAOUT, 20-1
Source program, 20-24, 20-26
Working storage, 20-26
Program communication block
PCB, 3-23
Program specification block
PSB, 3-23
PSB, 3-23

## Q

Qualifiers, 5-5, E-17

## R

R06FIGSA, D-31
R06PREPA, D-5
R13FIGSA, D-98
RA statement, E-17
Range category, E-18
Range graph, 14-31
RATIO, 14-20
Raw Data Output, 7-25
RC statement, E-18
Record, E-18
Record compression, 3-90
Record control field, 3-92
Record format, E-18
Record size, E-18
Record structure
VSAM, 3-21
RECORDSIZE parameter, 3-21
Referential integrity, 3-44, 15-49, 15-50, 15-51, 15-52, E-18

Designing for, 15-51
Maintaining values, 15-51, 15-52
Reject file, 10-1, 10-5, E-18
Rejected transaction file, 2-17, 7-22, 10-5
Definition of, 2-17
Relational expression, E-18
Relational files
Deletion of rows, 15-55, 15-56
Insertion of rows, 15-52, 15-53, 15-54, 15-55
Order of updating, 15-56, 15-57
Referential integrity, 15-49, 15-50, 15-51, 15-52
Set operations, 15-45, 15-46, 15-47, 15-48
Updating considerations, 15-42, 15-44, 15-45, 15-46, 15-47, 15-48, 15-49, 15-50, 15-51, 15-52, 15-53, 15-54, 15-55, 15-56, 15-57
Relational operator, E-18
Relational support, 16-12, 16-13
Foreign keys, 16-16
Uniqueness of rows, 16-16
Validation against DB2 catalog, 16-16
Relational table
To subfile, 7-11, 7-12
RELEASE, 8-21, 8-23, 8-24
Repeating subtitles, 9-14
REPLACE (P), 10-17
REPLACE (R), 10-17
Replace operations, 4-13
Replacement operator, E-19
Report, 2-21, 2-22, 5-5, 5-9, 14-1, 14-2, 14-3, 14-4, 14-5, 14-6, 14-7, 14-8, 14-9, 14-10, 14-11, 14-13, 14-16, 14-17, 14-18, 14-19
Alternate file, 7-5
Auto grand summaries, 9-13
Calculations, 14-20
Collating, 17-1
Concepts, 9-1
Control break field, 9-8
Customize decimal places, 14-20
Defaults, 9-4
Dynamic report line modification, 14-19, 14-20, 14-21, 14-22, 14-23, 14-24, 14-25, 14-26
Files, 7-4, 7-5
Flags, 14-4, 14-14
Formatted reporting, 14-13, 14-16, 14-17, 14-18, 14-19
Formatted sectional, 14-1, 14-2, 14-3, 14-4, 14-5, 14-6, 14-7, 14-8, 14-9, 14-10, 14-11, 14-19
From arrays, 13-21, 13-22
Grouping data, 9-8
Headings, 9-4
Increase lines per page, 14-20
Literals, 14-4, 14-13
Multi-image, E-14
Multiple, 5-14, 5-15, 5-16
Multiple files, 17-12
Names of fields, 14-4, 14-13
Output edit, 9-4
Page subtitles, 9-13
Painting, 14-49
Post-processing, 7-14, 7-16, 7-18, 7-19
Procedures, 5-9
Processing optimization, 9-19
Produce, 5-8
Remote output, 17-1
Repeating subtitles, 9-14
Round output, 14-20
Routing, 17-1
Simulation, 2-3
Sort, 9-8, 9-9
Sort fields, 4-8, 9-8
Spacing, 9-7
Standard values, 7-4
Subtitles, 9-13, 9-14
Summarizing data, 9-11
Two column, 9-7
Using field output edit, 9-16
report
SINGLESPACE, 9-15
REPORT command, 9-13, 9-14
report command, 9-15
Report command group, 9-3, 9-9, 9-10, 9-18
Report file, 2-17, E-19

Definition of, 2-17
Report files, 17-3
Report key, 17-3
Report listings file, 7-23
Report manager
Collating key, 17-3
Considerations, 17-10
JCL considerations, 17-12
Multiple report files, 17-12
Overview, 17-1
Report files, 17-3
Report key, 17-3
Routing key, 17-4
Sort key, 17-3
Storage considerations, 17-13
Report phase, E-19
Report step, 5-16, 7-5
Report summary file, 2-17, 7-14, 7-16, 7-18, 7-19
Generated file definition, 7-18
Record format, 7-16
Reporting concepts, 9-1
Request, 11-5, 11-6, 11-7, 11-8, E-19
Branching, 11-5, 11-10, 11-11
Cataloging, 19-5
cataloging, 19-6
Sequencing, 11-4
Types, 11-4
Request chain, E-19
Request groups
cataloging, 19-5, 19-6
REQUEST request name branch, 11-8
Request type, E-19
Restart, E-4
Result
Operand C, E-19
RETURN branch, 11-8
Revalidation, 4-18
RF statement, E-19
RG statement, E-19
Rn statement, E-17
Rn statement (graphics), E-17
Root segment, E-19
ROUND, 14-20
Rounding, 3-17
Routing key, 17-4
Routing reports, 17-1
RP statement, E-20
RRDS, 3-21
RT statement, E-20
Run control group, E-20
Run data groups, 19-4
Run statistics, E-20

## S

Sample Application Source Listing, 5-18
Scale
Graphics, 14-31
Scan only run, E-20
Scan operators, E-20
Scan pattern, E-20
Scatter diagram, 14-27, 14-29, 14-32, 14-34, 14-35, 14-37, 14-46, E-20
Floating point arithmetic, 14-46
Linear fit, 14-35
Linear prediction, 14-37
No fit, 14-33
Plotted points, 14-46
Summary page, 14-36
Summary statistics, 14-37
Scientific notation, 4-3
Secondary indexing, 3-32, 3-33, 16-10
Inverted structure, 3-32
Preselection, 3-33
Virtual key, 3-34
Segment, 2-3, 3-3, 3-5, 3-6, 3-8, 8-1, 10-10, E-20
Define, 3-13
Information, 3-9

Key, 3-2, 3-6
Level, 3-3, 3-6
Name, 3-15
Occurrences, 3-6
Transaction, 10-16
Type, 3-6
Segment functions, E-21
Segment key, E-21
Segment level, E-21
Segment name, E-21
Segment number, E-21
Segment occurrence, E-21
Segment operators, 8-21
Segment type, E-21
SELECT statement, 16-14, 16-15, E-21
Column functions, 16-15
Selection control, 8-20, 8-21, E-21
Sequence checking, 3-31, 6-5, E-21
Sequential access, 6-5
Sequential coordination, 4-19, 6-17, 6-18, 6-19, 6-20, 6-21, 6-22, 6-23, 6-24, 6-26, 6-27, 6-28, 6-45, 6-46
Multiple file, 6-29, 6-30, 6-31, 6-32, 6-33, 6-34, 6-35, 6-36, 6-37, 6-38, 6-39, 6-40, 6-41, 6-43, 6-44
Rules, 6-45
Single file, 6-21
Skip extra, 6-20
Terminate, 6-20
Transaction processing, 6-45, 6-46
Sequential file coordination cycle, E-22
Sequential table, 12-3, E-22
Siblings, E-22
Significance loss, E-22
Simulation, 2-3
Single path, 3-3
Single-path nested loops, 8-5, 8-6, 8-7, 8-12, 8-13
Single-step processing, 5-17, 9-19
SKIP, 14-20
Skip extra coordination, 6-20
Sort
Control field, E-22
Fields, 4-8, 9-8, 9-9
Sort key, 17-3
Sort phase, E-22
Sort step, 5-16
Source statement listing, E-22
Source statements, 6-59
SPACEAFTER, 14-21
SPACEBEFORE, 14-21
SSP, 9-19
Standard processing, 16-8, 16-9
Example, 16-8, 16-9
IMS, 16-3
Standard reports, 2-21
Standard subtitle, E-22
Start search, 6-6
Start/end search, E-23
Statement, E-23
Status indicator
Incomplete, 7-18
Invalid, 7-18
Not available, 7-17
Valid, 7-18
Step Including Transaction Procedures and Coordinated Files, 15-3
Sterling Library Utilities, 2-1, 2-4
Storage considerations, 16-11
Dynamic report line modification, 14-23
Storage requirements
Reducing, 16-11
Storage statistics, E-23
Storage utilization, 16-11
STRAN flag, 15-12
Structured files, 3-4, 3-5, 3-6
Defining, 3-14
Important points, 3-5
Structured record, E-23

SUB request name branch, 11-8
SUB set name branch, 11-8
Subfile, 2-17, 4-8, 5-5, 7-6, 7-7, 7-8, 7-9
  Calling, 7-7
  Considerations, 16-4, 16-5
  Define, 7-6, 7-7
  Definition of, 2-17
  Entire record selection, 7-13
  Format, 5-8, 7-8
  From an array, 13-27
  IMS databases, 7-10
  Multiple output to, 7-10
  Output to, 7-9
  Output to direct-access data sets, 7-10
  Removing dynamically deleted segments, 7-13
  Selected fields, 7-8
  To relational table, 7-11, 7-12
  Update sequentially coordinated file, 7-10
Subfiles, 7-6, E-23
Subroutine, 5-8
  Branching, 11-8
  Inherit, looping, 8-23
  Looping, 8-4, 8-24, 16-6
  Procedures, E-23
  Request set, 11-11, E-23
  Requests, 11-11, 11-13, E-23
  Sets, 11-13
Subtitle, E-23
Subtitles, repeating, 9-14
SUBTRACT FROM, 10-17
Summaries, E-23
Summarizing report data, 9-11
Summary axis, E-23
Summary labels, 4-14, 9-12, 9-13, 9-16, E-24
Summary level, E-24
Summary line, E-24
Summary only report, E-24
Summary section, 14-3, 14-6, 14-13
SUMMARY_ONLY, 9-14
Suppress
  Printing, 14-19
SUPPRESS SPACE, 14-20

## T

Tab-Delimited File, 2-17, 7-25, 7-26
Table, 12-1, E-24
  Argument, E-24
  Binary, 12-2, 12-3
  Define, 2-3, 12-1, 12-5, 12-6
  Definition of, 12-1
  Displacement, 12-4, 12-5
  Hierarchically organized keys, 3-79
  Lookup, 2-20, 4-8, 12-6
  Lookup automatic, 12-6, 12-7
  Lookup procedural, 12-7, 12-8
  Normalized, 3-63
  Result, E-24
  Search - advanced, 12-9
  Search - equal, 12-8, 12-9
  Search - interpolation, 12-13, 12-14, 12-15
  Search - larger, 12-11
  Search - nearest, 12-10
  Search - out of range, 12-15
  Search - smaller, 12-12
  Sequential, 12-3
  Types, 12-2
Table lookup, E-24
  Automatic, E-2
  Definition of, 12-1
  Linear interpolation, E-13
  Procedural, E-17
TB statement, E-25
TD statement, E-25
TE statement, E-25
technical support, contacting Computer Associates, 1-6
Temporary fields, 4-16, 4-17, 4-18, 5-6, E-25
  Default attributes, 4-17
  Definition of, 4-16

Explicit, 4-17
Initial value, 4-17
Working storage, 3-93
TF statement, E-25
Time data
Formats, 4-9
Time fields, 4-3, 4-8, 4-9, E-25
TITLE command, 9-3
TITLE LINENO, 14-20
TL statement, E-25
Tn statement, E-24
Tn statement (graphics), E-24
Top of a request, 11-10
Top of request, E-25
Total, E-25
Grand, 9-13
Summary, 9-11
Trailer group, E-25
Transaction
Action codes, 4-6
Cycle, 10-5, 10-6, 10-7
Definitions, 2-2, 10-1, 10-7, 10-10, 10-13, 10-14, 10-15, 10-16, E-25
DELETE flag, 10-16
Direct-read master file, 10-6
ECORD flag, 15-15
IMS considerations, 15-42
LSTATUS flag, 15-15
Sequential master file, 10-5
STRAN flag, 15-12
TRAN flag, 15-14
Type 1, 15-6
Type 2, 15-6
Type 3, 15-7
Type 4, 15-7
Type M, 15-7
Update-in-place, 10-6
XTRAN flag, 15-12
Transaction cycle, E-25
Transaction file, 2-15, 6-2, 6-9, 7-22, 10-1, 10-6, 10-7, 10-10, E-26
Definition of, 2-15
Eliminate duplicate keys, 15-41
Header/trailer processing, 15-41
Hierarchically structured, 15-41
Pre-summarization, 15-40
Rejected, 7-22
Uses other than updating, 15-39
Transaction group, 10-10, E-26
Transaction identifier, E-26
Transaction master technique, 15-39, 15-40, 15-41
Transaction procedures, 15-3, 15-6, 15-8, 15-9, 15-10, 15-11, 15-12, 15-13, 15-14, 15-15, 15-16, 15-17, 15-19, 15-22
Alignment, 15-13, 15-14, 15-15
Identification, 15-12, 15-13
Rejection, 15-16, 15-17
Type, 15-6
Type 1, 15-9, 15-12, 15-13
Type 2, 15-10, 15-13, 15-14, 15-15
Type 3, 15-11, 15-12, 15-13, 15-15, 15-16
Type 4, 15-12, 15-16, 15-17
Type M, 15-11, 15-15, 15-16
Update, 15-17, 15-19, 15-22
Updating, 15-15, 15-16
Validation, 15-12, 15-13
Transaction processing, 2-2, 2-18, 4-6, 4-11, 10-1, 10-4, 10-5, 10-6, 10-7, 10-16, 15-1, 15-3, 15-6, 15-8, 15-9, 15-10, 15-11, 15-12, 15-13, 15-14, 15-15, 15-16, 15-17, 15-19, 15-22, E-26
Flow, 10-3
Primary files, 15-8
Transaction processing cycle, 6-45, 6-46
Transaction record, E-26
Transaction requests, E-26
Transaction type 0, 15-6
Transaction type 5, 15-7
TRANSFER TO NEXT_MASTER, 5-13, 6-20
Tuning
Program analyzer, 20-1
Type 0 procedures, 15-6

Type 1 procedures, 11-8, 15-6
Type 2 procedures, 11-8
Type 3 procedures, 15-7
Type 4 procedures, 15-7
Type 5 procedures, 15-7
Type A requests, 11-4
Type E requests, 11-4
Type M procedures, 15-7
Type N requests, 11-4
Type S requests, 11-4

## U

UC statement, E-26
Unconditional branch, 11-5
UNMATCH label, 15-14
UPDATE statement, 16-15, 16-16
Update-in-place, 6-6, 7-3, 10-6
Updating
  Considerations, 16-3, 16-4, 16-5
  MOSAIC processing, 16-6
  Transaction processing, 6-45, 6-46
User coordination, 6-46, 6-47, 6-48, 6-49, 6-50, 6-51, 6-52, 6-53, 6-54, 6-55, 6-56, 6-57, 6-58, 6-59
  Direct-read, 6-47, 6-48
  Examples, 6-48, 6-49, 6-50, 6-51, 6-52, 6-53, 6-54, 6-55, 6-56, 6-57, 6-58, 6-59
  Fixed format listing, 6-59
  Serial read, 6-47
User I/O, 6-3, 7-3
User-read coordination, E-26

## V

Variable length fields, 3-5, 4-5, 4-6, E-27
  Formatted reporting, 14-18
  In reports, 4-6
  Output edit length, 4-14
  Segments, 4-5
  Temporary, 4-5
  Transaction action codes, 4-6
Variable length segment, 3-6
Variable occurring segment, 3-6, 3-8, E-27
Virtual key, 3-34, E-27
Virtual segments, E-27
VISION:Builder
  Calls to an external program, 18-8
  Engine, 2-1
  Linkage section, 18-1, 18-2, 18-3, 18-5, 18-6, 18-8, 18-9, 18-10, 18-11, 18-12, 18-14, 18-15, 18-16, 18-18, 18-20, 18-22, 18-23
  Specifying external calls, 18-8, 18-9, 18-10, 18-11
  Type, 1-6
VISION:Builder system, 2-1
VISION:Clearaccess Reports File, 2-17
VISION:Workbench for DOS, 2-1, 2-2, 3-1, 3-6, 3-9
  Array Definition window, 13-4
  Column Heading window, 3-20
  File Definition window, 3-6, 3-14, 3-15
  File List window, 3-10
  Output Edit window, 3-17
  Segment Definition window, 3-15, 4-15, 4-16, 9-2
VISION:Workbench for ISPF, 2-1, 2-3, 3-1
VSAM
  Alternate indexing, 3-22
  Cluster definition, 3-21
  Define a file, 3-20
  Record structure, 3-21
VSE
  Linking to VISION:Builder, 18-4

## W

web page
  Computer Associates, 1-6
WH statement, 3-70, 3-71, 3-73, 3-74, 3-75, 3-76, 3-77, 3-78, 3-79, 6-8, 16-13, 16-14, E-27
Work areas, 5-5, 10-5, 10-7, E-27
  Processing, 5-4, 5-5
Working storage, 3-93, 5-6, E-27

Example, 15-17
Program analyzer, 20-26

## X

XDFLD SRCH, 3-32
XTRAN flag, 15-13

## Z

Zoned decimal fields, 4-3
Output edit length, 4-15