

Advantage™ VISION:Workbench™ for DOS

Reference Guide

6.0



Computer Associates™

WDUSR060.PDF/D50-201-011

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

VISION:Workbench Structure	1-2
Main Menu	1-2
Application Development.....	1-2
Definitions	1-2
Miscellaneous	1-2
Organization of This Book	1-3

Chapter 2: Getting Started – Installation and Setup

Installing VISION:Workbench for DOS	2-1
System Requirements	2-1
Compact Disc	2-2
Installation Instructions.....	2-2
Starting VISION:Workbench for DOS.....	2-3
Specifying Setup Information.....	2-4
System Parameters	2-7
Displaying System Parameters.....	2-7
Changing System Parameters.....	2-7

Chapter 3: VISION:Workbench Concepts

Requests and Procedures	3-1
Advanced Syntax Language (ASL).....	3-1
File Structures in Pictures.....	3-2
Orphans	3-2
VISION:Builder Processing Control Procedures	3-2
VISION:Builder Processing Objects.....	3-4
VISION:Builder Processing Transactions	3-5
VISION:Transact Screen Control.....	3-6

Chapter 4: Using the System

Navigation.....	4-1
Navigating from Menus.....	4-2
Navigating from List Windows.....	4-3
Navigating from Definition Charts	4-4
Navigating Upwards	4-5
Function Keys	4-5
Switch (Alt+F8).....	4-7
Button Fields	4-7
Entering Text.....	4-7
Entry Descriptions	4-8
Help Information.....	4-9
Workbench Prompts	4-10
Using the Editing Menu Bar Items	4-11
Documenting Applications.....	4-13
Modifying the Application	4-15
Adding Text Lines	4-15
Inserting Text Lines.....	4-16
Deleting Text Lines.....	4-16
Validating Applications.....	4-17
Saving Applications	4-18
Deleting Applications and Application Objects.....	4-19

Chapter 5: Entering Definitions

Definition Component.....	5-1
Definition Lists	5-2
File Definitions.....	5-2
Table Definitions.....	5-4
Array Definitions.....	5-6
Transaction Definitions.....	5-7
Defining Files	5-9
Using the Segment Structure Chart	5-9
Navigating Through the Segment Chart	5-10
Defining a New File	5-11
Deleting Segments from the Structure	5-14

Definition Menu Bar Items	5-14
Using RELATION.....	5-16
Using ABOVE	5-16
Using ORPHANS	5-17
Using RETRIEVE.....	5-17
Using DELETE.....	5-17
Using EXPORT.....	5-17
Using EDIT.....	5-17
Using CENTER.....	5-18
Defining VISION:Builder Transactions.....	5-18
Defining the Transaction Group.....	5-19
Defining the Transaction	5-20

Chapter 6: Creating VISION:Builder Applications

Application Development Windows.....	6-1
VISION:Builder Application List Window.....	6-1
Application Definition Window	6-2
Files Used Window	6-5
Application Parameters Window.....	6-6
Transaction Groups Used Window.....	6-6
Application Comments Window	6-7
Subfile Definition Window	6-7
Procedure Definition Window	6-9
Request Definition Window	6-10
Report Definition Window	6-12
Report Manager Window.....	6-13
Retrieving Application Objects	6-14

Chapter 7: Creating Reports

Report Definition Windows.....	7-1
Report Definition Window	7-2
Report Preface Window.....	7-3
Report Details Window	7-4
Report Column Headings Window.....	7-5
Report Comments Window	7-6
Report Title Window.....	7-7
Report Extended Titles Window	7-8
Report Summaries Window	7-9

Using the Report Definition Menu Bar	7-10
GEN Menu Bar Item	7-10
CONTENTS Menu Bar Item	7-10
LEVEL Menu Bar Item	7-10
SIMULATE Menu Bar Item.....	7-10
Simulating Reports.....	7-11

Chapter 8: Walk-Through for VISION:Builder Applications

Application Overview	8-1
Important Keys	8-2
Beginning the Application	8-2

Chapter 9: Creating VISION:Transact Applications

Free-Form Language.....	9-5
Summary of Validation Statements and Output Conditions.....	9-5
Application Development Windows.....	9-6
VISION:Transact Input Control Procedures.....	9-7
VISION:Transact Processing Objects.....	9-8
VISION:Transact Application List	9-8
Application Definition Window	9-9
Application Definition Menu Bar Items	9-10
Using RETRIEVE.....	9-10
Using EXPORT.....	9-10
Using Prototype.....	9-10
Files Used Windows	9-12
Screen Flow Window	9-13
VISION:Transact Screen Layout.....	9-14
VISION:Transact Screen Definition	9-14
Screen Flow Menu Bar Items	9-14
Using ABOVE	9-15
Using SCREEN	9-16
Using PROTOTYPE	9-17
Using RETRIEVE.....	9-17
Conditions Window	9-17
Selecting a Follow-on Screen	9-19
Application Environment Window	9-20
Application Comments Window	9-21
Screen Definition Window	9-21
Procedure Definition Window	9-22
Request Definition Window	9-23

Chapter 10: Screen Layout, Contents, and Actions

Screen Layout.....	10-3
Screen Layout Menu Bar Items	10-4
Using CENTER.....	10-4
Using EDIT	10-4
Using ATTRIBUTES.....	10-5
Using PAINT	10-6
Using CURSOR.....	10-6
Contents Window.....	10-7
Additional Validation/ Actions	10-8

Chapter 11: Walk-Through for VISION:Transact Applications

Application Overview	11-2
Important Keys	11-5
Beginning the Application.....	11-6

Chapter 12: Utilities and Host Activity

Utilities.....	12-1
Print Library Items Utility	12-2
ALL, SELECTIONS, and NAMES Menu Bar Items	12-3
Importing Applications and Definitions	12-4
Transaction Definition Import Considerations	12-9
Application Import Considerations.....	12-10
Exporting to the Host	12-10
What EXPORT Does.....	12-11
About EXPORTing Files	12-12

Appendix A: A VISION:Builder User's Guide to VISION:Workbench

Naming Conventions.....	A-1
Comment (AA) Statement	A-1
Establish Request (ER) Statement.....	A-2
Procedural (PR) Statement.....	A-2
Run Control (RC) Statement.....	A-2
Application Files (RF) Statement	A-3
Run Parameter (RP) Statement.....	A-3
Temporary Field (TF) Statement	A-3

Appendix B: A VISION:Transact User's Guide to VISION:Workbench

Naming Conventions.....	B-1
Comment (AA) Statement	B-2
Establish Request (ER) Statement	B-2
Procedural (PR) Statement.....	B-3
Run Control (RC) Statement.....	B-3
Application Files (RF) Statement	B-3
Run Parameter (RP) Statement.....	B-3
Screen Control (SC) Statement	B-4
Screen Definition (SD) Statement.....	B-5
Screen Layout (SL) Statement.....	B-5
Screen Field (S0) and Screen Content (SN) Statements	B-6
Screen Segment (SS) Statement	B-8
Temporary Field (TF) Statement	B-8

Index

Introduction

VISION:Workbench™ for DOS is an application development system executing on IBM® personal computers and compatibles. With VISION:Workbench, you can develop host applications right on your own PC. It gives you host application development combined with the ease of use, flexibility, and responsiveness of the desktop computer.

In addition, you enjoy instantaneous response to entries along with their immediate validation. There is no waiting for a response from a remote computer.

Major features include:

- Structured application design.
- Procedural language.
- Interactive error validation.
- Function key directed actions.
- Orphan creation and retrieval.
- Multi-window display.
- PC simulation of host processing.
- Menu-selected functions.
- Online help information.
- Text editing functions.
- Graphically represented data definitions.
- Item name display lists.
- Built-in documentation facilities.
- Import/export utilities.

The following features are specific to the VISION:Transact® portion of the system:

- Graphically represented application flow.
- Screen painting.
- Interactive prototyping.

The Report simulation feature is specific to the VISION:Builder® portion of the system.

VISION:Workbench Structure

VISION:Workbench is a menu-driven system. Each of its major components is accessed through the Main Menu. You access one of the development components or the miscellaneous component by selecting it on the Main Menu. [Figure 1-1](#) shows the Main Menu.

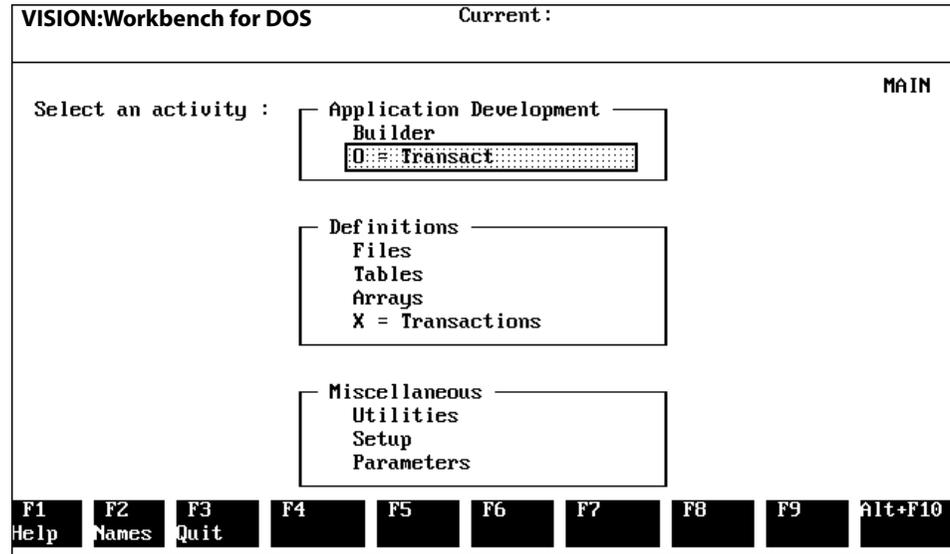


Figure 1-1 VISION:Workbench for DOS Main Menu

Main Menu

The Main Menu displays when VISION:Workbench is invoked. It contains a list of the major components of the system. You select one of these major components from the list. From this menu, you can access all components of the system.

Application Development

This menu is used to create, modify, or delete VISION:Builder (Builder) and VISION:Transact (Transact) applications. From this menu, you can access the other Builder and Transact application windows.

Definitions

This menu is used to create, modify, or delete file definitions, transaction definitions, table definitions, or array definitions. From this menu, you can access the other definition windows.

Miscellaneous

This menu is used to invoke utilities and to specify and modify VISION:Workbench setup and parameters.

Organization of This Book

This book contains information specific to the operation and use of VISION:Workbench. It is designed to give you a basic understanding of the features, capabilities, and flow of VISION:Workbench.

The following list describes the contents of the chapters of this book:

- [Chapter 1, *Introduction*](#) introduces VISION:Workbench, provides an overview of its operation, and describes the contents of this book.
- [Chapter 2, *Getting Started – Installation and Setup*](#) describes the installation procedures for VISION:Workbench for DOS.
- [Chapter 3, *VISION:Workbench Concepts*](#) introduces several concepts in the Application Development process. These concepts are discussed throughout this document but are grouped together in this chapter for ease of reference.
- [Chapter 4, *Using the System*](#) contains information about the text entry windows. It describes important information for entering and editing text on VISION:Workbench windows. It also describes function keys, the menu bar, and help information, as well as documenting, saving, and validating applications and application objects.
- [Chapter 5, *Entering Definitions*](#) describes the definition component of the system. It contains information about the windows in the definition portion of the system and the menu bar items and their functions. Important entries for defining files and the use of orphans is discussed. For VISION:Builder applications, defining tables, arrays, and transactions are also discussed.
- [Chapter 6, *Creating VISION:Builder Applications*](#) describes the application component windows in the Builder portion of the system. It also contains information about specifying files, parameters, transaction groups, reports, procedures, and subfiles, and how to retrieve reports, subfiles, and procedures.
- [Chapter 7, *Creating Reports*](#) describes VISION:Builder report creation. It contains information about the report windows, important report entries, the various elements in a report (such as titles and column headings), summary reports, alternate report files, formatted sectional reporting, and report simulation.

- [*Chapter 8, Walk-Through for VISION:Builder Applications*](#) takes you through a VISION:Builder application example to show you how easy the Builder portion of VISION:Workbench is to use. The example is not designed to teach you how to use VISION:Builder; it is designed as a walk-through of VISION:Workbench. Once you complete the example, you should be able to continue using VISION:Workbench to develop your own VISION:Builder applications.
- [*Chapter 9, Creating VISION:Transact Applications*](#) describes the creation of VISION:Transact applications.
- [*Chapter 10, Screen Layout, Contents, and Actions*](#) describes screen painting, contents, and actions for VISION:Transact applications.
- [*Chapter 11, Walk-Through for VISION:Transact Applications*](#) takes you through a VISION:Transact application example to show you how easy the Transact portion of VISION:Workbench is to use. The example is not designed to teach you how to use VISION:Transact; it is designed as a walk-through of VISION:Workbench. Once you complete the example, you should be able to continue using VISION:Workbench to develop your own VISION:Transact applications.
- [*Chapter 12, Utilities and Host Activity*](#) describes the system requirements, administrative functions (system setup and parameters), and the conversion utilities. It also contains information about sending VISION:Workbench generated applications to the host, downloading definitions, and special considerations for interfacing with the host.
- [*Appendix A, A VISION:Builder User's Guide to VISION:Workbench*](#) explains the VISION:Workbench for DOS in VISION:Builder fixed format terms.
- [*Appendix B, A VISION:Transact User's Guide to VISION:Workbench*](#) explains the VISION:Workbench for DOS in VISION:Transact fixed format terms.

Getting Started – Installation and Setup

Installing VISION:Workbench for DOS

The VISION:Workbench installation procedure is simple and straightforward. The system and sample applications are distributed on one compact disc. The installation procedure creates a directory and installs the program files and sample program to that directory.

VISION:Workbench installation consists of the following activities:

- Optionally setting the MPCPROF environment variable to point to your profile (setup) information. (This is useful when VISION:Workbench is installed on a LAN.)
- Tailoring VISION:Workbench to your environment. This involves starting VISION:Workbench and telling it, among other things, the paths and subdirectories where you want to keep VISION:Workbench for DOS applications, files to be imported, and files to be exported.

System Requirements

The minimum requirements to run the VISION:Workbench for DOS are as follows:

- An IBM PC/AT or compatible.
- MS-DOS (PC-DOS) version 3.0 or later.
- Two megabytes of RAM.
- A hard disk.
- A CD-ROM drive.
- A monochrome or color monitor.

A printer is optional.

VISION:Workbench for DOS does not require a third-party memory manager. However, if a third-party memory manager is resident, it must be VCPI (Virtual Control Program Interface) compliant. The following memory managers are known to conform to these requirements:

- QEMM 6.0+ from Quarterdeck
- 386MAX 6.0+ from Qalitas
- HIMEM 5.0 from Microsoft
- EMM386 6.0 from Microsoft (without the NOEMS parameter)

Note: EMM386 5.0 does not conform to VCPI requirements, and VISION:Workbench for DOS will not operate when it is resident even if HIMEM is also resident.

Compact Disc

The compact disc contains the VISION:Workbench program, related files, sample applications, and online documentation.

Installation Instructions

The following steps describe the installation procedure. (Substitute your own drive addresses in DOS commands used.) VISION:Workbench creates or replaces a directory named MPCxx (where xx is the release number) and a subdirectory under it named SAMPAPPS.

1. To install VISION:Workbench, insert the compact disc into the CD-ROM drive and enter a DOS command similar to the one below.

```
A:INSTALL A C
```

This command installs from the CD-ROM drive onto the hard disk drive C.

2. Remove the CD after you receive the message "Installation is complete!". You are now ready to use VISION:Workbench. See [Starting VISION:Workbench for DOS on page 2-3](#).

For LAN Users: The environment variable MPCPROF allows each user on a LAN to specify the path to be used in loading individual profile information, such as paths and subdirectories. The profile information is specified on the Setup window and, with the use of MPCPROF, can be different for each user.

File MPCPROF.TXT contains profile information. If the environment variable MPCPROF is found during startup, VISION:Workbench uses the assigned directory to load (or save) profile information. If MPCPROF is not found during startup, the current directory is used.

If you want your profile information saved in a directory other than the current directory, you must set MPCPROF prior to saving your profile information. MPCPROF is assigned using the SET statement as follows:

```
SET MPCPROF=drive:\directory
```

You can include this statement in your AUTOEXEC.BAT file where appropriate.

Starting VISION:Workbench for DOS

This section describes how you start VISION:Workbench. You do not have to worry about changing system parameters when you first start using the system, but you may have to change the default system delimiter if it does not match the one on your host. (See [Changing System Parameters on page 2-7](#) and [Displaying System Parameters on page 2-7](#). Also see [Specifying Setup Information on page 2-4](#).)

The following steps describe how to start VISION:Workbench:

1. Make the drive containing VISION:Workbench the default drive. For example, if this is drive C, type: `c:` and press Enter.
2. Make the directory containing VISION:Workbench the default directory. For example, if this directory is MPCxx (where xx is the release number), type: `cd \mpcxx` and press Enter.

You can enter an optional `-P` parameter specifying the path name for locating the profile information MPCPROF.TXT on the startup command. For example, `mpcx -p:\myprof`.

When present, the `-P` parameter supersedes the setting in the MPCPROF environment variable.

3. Invoke the VISION:Workbench for DOS system, type: `mpcx` and press Enter.
Or, specify a path name for locating the profile information MPCPROF.TXT. For example: `mpcx -P c:\myprof` and press Enter. When present, the `-P` parameter supersedes the setting in the MPCPROF environment variable.

Specifying Setup Information

The Workbench Setup window, shown completed in [Figure 2-1](#), contains the following entries:

- Library Path tells VISION:Workbench the path and directory where you want to store your applications.
- Import Input Path is the path and directory where you want to store applications and definitions to be imported to VISION:Workbench. The directory specified must have enough space to contain a temporary work file written for the imported application (the work file is automatically deleted at the end of the file import process).
- Export Output Path is path and directory where you want to store applications to be uploaded to the host.
- Printer Output Dest is the path and file name or device name for VISION:Workbench to use during the Print function.
- Printer Setup allows you to specify the capabilities of the printer. Press F8 to open an additional window with the following entries:
 - Printer Graphics Character Support indicates whether your printer can print the IBM extended graphics character set.
 - Print Initialization allows you to specify the string of characters to send to the printer prior to printing.
 - Print Termination allows you to specify the string of characters to send to the printer after printing.
 - Compression On allows you to enter the string of characters to send to the printer to initiate compressed printing mode.
 - Compression Off allows you to enter the string of characters to terminate compressed printing mode.

PC Display Monitor indicates the type of monitor attached to your system.

Target Mainframe Support Native ASL specifies whether your local host processor supports the ASL procedural language. If you are using VISION:Builder Release 11.0 or later, or VISION:Transact Release 6.0 or later, we recommend that you set this parameter to YES.

VISION:Workbench for DOS		Current :
Setup Specifications		US0100
Library Path :		
Primary	C:\MPCXX\SAMPAPPS	
Import Input Path:	C:\MPCXX\IMPORT	
Export Output Path:	C:\MPCXX\EXPORT	
Printer Output Dest:	PRN	
Printer Setup: Yes	PC Display Monitor: COLOR	
Target Mainframe Support		
Native ASL:	YES	
F1	F2	F3
Help		Close
F4	F5	F6
	Save	
F7	F8	F9
		Alt+F10
Path Name		

Figure 2-1 Completed Setup Window

To specify setup information

1. Move the cursor to Setup on the main menu and press F8 (or just press S).
2. Enter the path to your application subdirectory next to Primary Library Path: For example, type: \mpcxx\sampapps (where xx is the release number) and press Enter to move to the next entry.
3. Under Import Input Path, enter the path where you store applications and definitions to be imported to VISION:Workbench. The files in this path must contain the source statements retrieved from the host. You can temporarily override this entry on the Utilities window. For example, type: \mpcxx\import (where xx is the release number) and press Enter to move to the next entry.
4. Enter the path for saving applications to be uploaded to the host next to Export Output Path: For example, type: \mpcxx\export (where xx is the release number) and press Enter to move to the next entry.

Note: Be sure that all subdirectories have been created before file activity on them begins. If you have not yet created the subdirectories, you can press Alt+D to return to DOS and create them. The DOS EXIT command returns you to VISION:Workbench.

5. Enter the Printer Output Destination for VISION:Workbench to use during the Print function. If this entry is left blank, the printer output is routed to the standard printer. You can temporarily override this entry on the Utilities window.
6. Press F8 on the Printer Setup entry to display a pop-up window where you can enter printer setup information.
7. Enter the PC Display Monitor type if you want to change the default monitor type for some reason. This is usually unnecessary because VISION:Workbench defaults to the monitor type attached to your PC. You may want to use it if you have to change monitors in mid-session. To change the default, enter one of the following monitor types (or press F6 for a Choices list):
 - Blank - VISION:Workbench will select choice based on the monitor being used.
 - M - for monochrome
 - C - for CGA/EGA/VGA color
 - O - for other monitor types (useful for some laptop models)
8. Enter a Y if your host processor supports the ASL language. Enter N if it does not. This entry applies to the Export facility.
9. Be sure you press F5 to save your setup information. See the example of a completed setup window in [Figure 2-1](#). Press F3 to close the window and return to the Main Menu.

You can request Help at any time for a description of the entries on this window.

The setup information is saved (in file MPCPROF.TXT) in the current directory unless the environment variable MPCPROF has been set. (See the note for LAN Users in [Installation Instructions on page 2-2](#).)

System Parameters

This section describes how to display the system parameters and how to change them. VISION:Workbench is installed with a complete set of system parameters. These parameters provide default values for many of the entries on VISION:Workbench windows. [Figure 2-2](#) shows the Run Parameters window that contains the default system parameters.

VISION:Workbench for DOS		Current:	
Run Parameters			
Character Definitions		Delimiters	
Digit Select Char.: 9	Currency Char.: \$	System Delimiter...: #	
Zero Suppress Char.: Z	Minus Char....: -	Today Format Delim: /	
Check Protect Char.: *	Non-Edit Field: +	Isdate Delimiter...: -	
Decimal Char.....: .	Auto Grand....: NO	Julian Delimiter...: .	
Grouping Char.....: ,	Percent Char..: %	Time Delimiter.....: :	
Plus Char.....: +	Heading Char...: -		
Online Parameters		Size Definitions	
Field Designator.: _	Temp Defs Reqd: NO	Def Width of Page.: 132	
Hilite Designator: 0	Connector Reqd: NO	Page Height.....: 66	
Target Monitor...: CICS		M4LIST Width.....: 132	
Terminal Devices.: 3			
		Date Format: MMM DD, YYYY	
F1 Help	F2	F3 Close	F4 Menu Bar
F5 Save	F6	F7	F8 Open
F9 Insert	Alt+F10		
Digit Select Character			

Figure 2-2 Run Parameters Window

Displaying System Parameters

You display the system parameters starting on VISION:Workbench Main Menu by selecting the Parameters entry in the Miscellaneous box. When you select this entry, the Run Parameters window, shown in [Figure 2-2](#), opens. You can change any parameter to meet your specific needs.

Changing System Parameters

This section describes how to change the default system delimiter. You can follow the example described in this section for changing any of the system parameters.

The system delimiter is one of the parameters that VISION:Workbench uses to create definitions and applications. VISION:Builder and VISION:Transact on the host also have system delimiters in their run parameters.

Your host product expects definitions and applications that it processes to have the same system delimiter as the one specified in its run parameters. Therefore, to send definitions and applications to the host for processing, you must make sure that VISION:Workbench uses the same system delimiter as host VISION:Builder or VISION:Transact.

If the system delimiters do not agree, your application will not be processed correctly on the host. It is easy to change the system delimiter if you determine that they are different.

To change the system delimiter

1. Find out what system delimiter host VISION:Builder or VISION:Transact is using. (You can find this out by looking at a signon page.)
2. Once you know what the system delimiter should be, select Parameters on the Main Menu and press F8 (the Run Parameters window in [Figure 2-2](#)). The System Delimiter entry displays in the upper-right column on the Run Parameters window.
3. Change the system delimiter by moving your cursor to the System Delimiter entry and keying the host system delimiter over the default delimiter.
4. Repeat steps [2-3](#) to change any additional parameters.
5. When you are finished changing parameters, save the run parameters by pressing F5. Press F3 to close the window and return to the Main Menu.

VISION:Workbench Concepts

VISION:Workbench introduces several new concepts in the application development process. These concepts are discussed throughout this document but are grouped together in this chapter for ease of reference.

Requests and Procedures

With VISION:Workbench, you can enter procedural statements in one of two ways. You can write procedural statements using the Advanced Syntax Language (ASL). When you use ASL, the statements are grouped into a set (object) known as a Proc.

You can also write procedural statements in a fixed format language familiar to VISION:Builder and VISION:Transact programmers. Statements written in the fixed format syntax are grouped together as an object known as a REQUEST.

Procs and REQUESTs can both be included in the same application; however, fixed format statements and ASL statements cannot appear in the same object. Procs contain ASL statements; REQUESTs contain fixed format statements.

Advanced Syntax Language (ASL)

A free-form 4GL language is supported by VISION:Workbench for DOS. This language is known as the Advanced Syntax Language or ASL. The language is fully discussed in the ASL Reference Guide.

ASL is directly supported in its native free-form syntax when using VISION:Workbench. When using the VISION:Builder or VISION:Transact engine, ASL is directly supported in its native free-form syntax in the OS/390® (MVS®) and CMS environments. Native ASL syntax is not directly supported by the engines in the VSE environment.

For VISION:Builder and VISION:Transact users in the VSE environment, VISION:Workbench will translate your native ASL syntax coding into fixed format coding during the export process. See VISION:Workbench Setup window description in [Chapter 2, Getting Started – Installation and Setup](#) for details on specifying target host support of native ASL.

File Structures in Pictures

File structures and conditions of joins between relational tables are specified pictorially. You draw the structure (relationship). [Chapter 5, Entering Definitions](#) discusses this process.

Orphans

Orphans are new objects introduced in VISION:Workbench. When a segment is deleted from a file definition, it disappears from the window. However, the segment (and all of its associated field definitions) has not been physically deleted from the system. It is still available. Because it is a segment with no parent, it is an orphan. Orphans can be retrieved, selected, and attached to a parent segment. Once attached, they cease to be orphans.

The concept of orphans greatly facilitates the reorganization of definitions. This is particularly valuable with relational tables, because each table is represented by a segment. File definitions are logical views (that is, they do not physically exist) and it is frequently desirable to construct several different logical views. This is easily achieved by selecting the orphans and establishing the appropriate relationships.

This same concept exists in VISION:Transact applications with screens and screen flow. Screen flow depicts the interaction between screens. A screen that is defined but not “hooked into” the screen flow is an orphan screen. It exists but is not part of the application. It becomes part of the application when it is connected into the screen flow.

VISION:Builder Processing Control Procedures

VISION:Builder applications assume a basic application cycle derived from the inputs and outputs defined for the application. For example, an application follows a fixed sequence of events:

- Read a master file record.
- Apply each transaction in turn.
- Process data after each VISION:Builder transaction.

The application cycle of reading records, matching files, writing records, and so on, takes place automatically. VISION:Workbench provides certain control points where you can insert a series of procedures/requests/report objects, in effect inserting customized application code superimposed on the automatic cycle. These control points occur in the cycle when certain events occur and your code is inserted as an event controlled procedure (that is, a procedure that is invoked when a certain event occurs).

The particular events where your code is executed are:

- After each master file record is read or updated.
- After each set of input files is synchronized (or aligned).
- When all files have reached end of file.

If you have a transaction file, there are some additional events:

- After reading a transaction record (for validation of the transaction record).
- After synchronizing the transaction record with the master file record.
- After updating the master file record with the transaction record.
- After updating the master file record but providing access to the entire record.
- When a transaction error is detected.

Only one event controlled procedure can be provided for each event. Event controlled procedures are identified by a window that displays when the application is initially identified. This is illustrated in [Figure 3-1](#).



Figure 3-1 Event Controlled Procedures Window

Event controlled procedures are responsible for activating other objects (procedures, reports, subfiles) in whatever order is wanted.

An event controlled procedure is a procedure that uses ASL. You activate other objects by issuing CALL statements.

While an event controlled procedure is a procedure written in ASL, you can call other procedures or conventional fixed format requests in any order. In effect, every object in the application can be thought of as subroutine (with the exception of the eight event controlled procedures).

An event controlled procedure is a procedure and is not restricted just to CALL statements. For simplicity in maintaining your application, keep the event controlled procedures as simple as possible. Where possible, restrict them to CALL statements governed by simple logic statements.

You only have to define event controlled procedures for the events that are of interest to you. At least one event controlled procedure must exist.

VISION:Builder Processing Objects

In VISION:Workbench for DOS, VISION:Builder applications are constructed by assembling together different objects. The following types of objects exist for VISION:Builder applications:

- PROC – A procedure written using ASL.
- REQUEST – A procedure written using fixed format syntax.
- REPORT – A report definition.
- SUBFILE – A subfile definition.

In contrast to VISION:Builder fixed format input, in all applications developed in VISION:Workbench for DOS, the objects are distinct and separate. You cannot define a subfile and a report in the same object; each is a separate definition. You cannot add report statements to the end of procedural statements; each is a separate object.

Applications are constructed from the event controlled procedures. From an event controlled procedure, you call other procedures, requests, reports, or subfiles as appropriate.

Again, all objects can be regarded as subroutines (except for the event controlled procedures). These subroutines can call another object and so on.

VISION:Builder Processing Transactions

Transactions are defined differently using VISION:Workbench than they are using VISION:Builder fixed format. In VISION:Workbench, a transaction is the interaction of one file (the transaction file) with another file (the master file). Because all files in VISION:Workbench are objects, they must have a file definition. Transaction files are files and, as such, must have a file definition. All fields in the transaction file that are used in the transaction definition must be defined (that is, they must have a field name). This includes the field that is used as the transaction identifier.

A transaction definition using fixed format is a mixture of file definition information (such as location, length, and type), transaction field data validation, and actions to be performed. In VISION:Workbench, a transaction definition consists only of defining the actions of the transaction file against the master file and the validation of the transaction data.

[Figure 3-2](#) illustrates an example of a transaction definition. Notice that all field references (particularly for the transaction file) are by field name, including the transaction identifier. Transaction definitions are discussed in [Chapter 5, Entering Definitions](#).

VISION:Workbench for DOS		Working :ORDENTRY	
COPY	MOVE	SEARCH	REPLACE
		Transaction Group:ORDENTRY	FD0250
		Master File.....:CUSTMAST	
		Transaction File.:TRANWORK	
ID name:ADDORDR	Description:ADD ORDER TRANSACTION		
		Name	Oper Code
Transaction Identifier No. 1 :	TRANID1	EQ	ADD
AND Transaction Identifier No. 2 :	TRANID2	EQ	ORDR
Transaction Field	Action Code	Master Field	Validation Type
CUSTNO	MATCH WITH	CUSTNO	PATTERN Y9999
ORDERNO	INSERT	ORDERNO	
ORDRDATE	XFER DATA ONLY	ORDRDATE	
ORPERSON	XFER DATA ONLY	ORPERSON	PATTERN -9-9-9
ORDPONUM	XFER DATA ONLY	ORDPONUM	
ORDDUDAT	XFER DATA ONLY	ORDDUDAT	
ORDINUGN	XFER DATA ONLY	ORDINUGN	PATTERN Z
ORDCMPLT	XFER DATA ONLY	ORDCMPLT	
Validation Criteria			
Pattern or Min - Max			
Required name			

Figure 3-2 A Transaction Definition

VISION:Transact Screen Control

The flow of screens within a VISION:Transact application is specified pictorially as a hierarchical structure, with follow-on screens represented as *children* of the screen that precedes them. You can specify a set of input control procedures, corresponding to VISION:Transact type 1, type 2, and type M requests, for each input screen. Associated with each output screen are conditions under which the screen is output, together with optional output control information (including the name of a procedure to be invoked prior to output, the equivalent of a VISION:Transact type O request). [Chapter 9, *Creating VISION:Transact Applications*](#) describes these specifications in greater detail.

Navigation

VISION:Workbench for DOS is easy to use. Its design consists of a series of windows that are organized in a hierarchical fashion. Navigation throughout VISION:Workbench structure is bidirectional — you either go up the structure or down the structure. You can encounter three types of windows in VISION:Workbench:

- Menus
- Lists
- Charts

Menus, lists, and charts are augmented by function keys and button fields, which are discussed in a later section.

Navigating from Menu

You always begin VISION:Workbench at the Main Menu shown in [Figure 4-1](#). The Main Menu lists the major components. When you select a component from this list, the appropriate window opens. To invoke any other major component, you simply proceed up the structure (pressing F3) until you return to the Main Menu, where you select the new component.

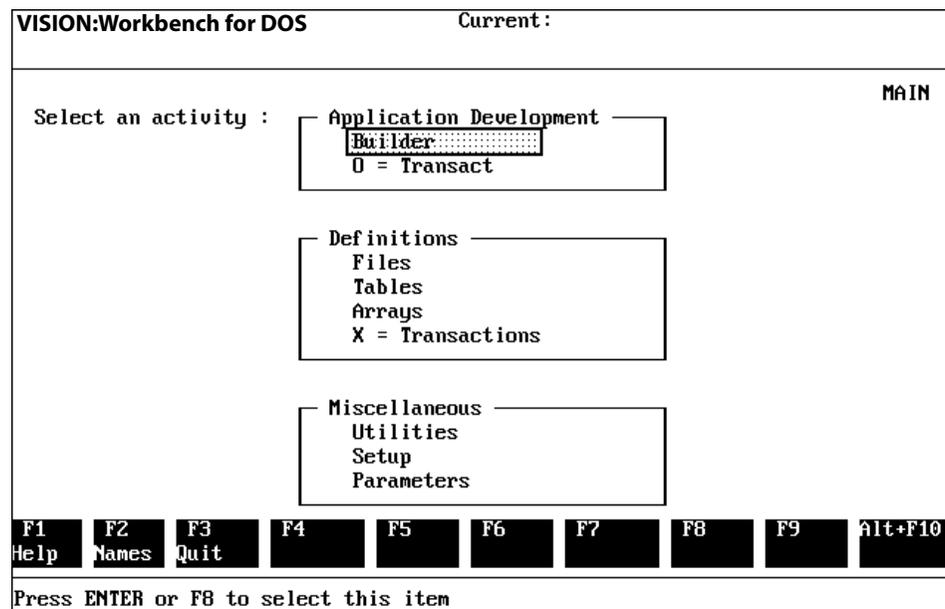


Figure 4-1 Navigating from the Main Menu

You select a component on the Main Menu, or any subsequent menu, in one of the following ways:

- Move your cursor to the name of the component that you want to invoke and press Enter or F8.
- Enter the first letter of the name of the component that you want to invoke from the menu list.

When you enter an F on the Main Menu, the File Definition List window opens. You can achieve the same results by moving the cursor to the Files entry in the list and pressing Enter or F8. From the Main Menu, as displayed, note that T invokes Table definitions while X invokes Transaction definitions.

Navigating from List Windows

Figure 4-2 shows a list window. To move to the next lower window following this window, place your cursor over the item that you want to select and press F8.

VISION:Workbench for DOS		Working:EX1	
X=EXPORT RETRIEVE			
Application Name:EX1			RC4000
Builder Application Sections			
Files.....:	1	Control	Params
Report Mgr:No	Procs....:	1	Requests: 0
			Transact: 0
			Comments: No
			Reports.: 1
			Subfiles: 0
Type	Name	Description	
PROC	VALUE	;EACH MASTER-FILE RECORD CONTROL PROCEDU	
REPORT	VALUEREP	REPORT SHOWING VALUE OF INVENTORY	
Name of item; press F8 to select this item.			

Figure 4-2 Navigating from a List Window

Using Figure 4-2 as an example, selecting the procedure VALUE opens the VALUE Procedure Definition window in Figure 4-3.

VISION:Workbench for DOS		Working:EX1	
COPY MOVE SEARCH REPLACE			
Procedure Name:	VALUE	Type : N	RQ0100
Pre-Selection :	N	Re-init Temps:	Maximum Items Selected:
;EACH MASTER-FILE RECORD CONTROL PROCEDURE			
;			
STOCKVAL: FIELD P 5 DEC 2			
IF QANONHND > 0			
LET T.STOCKVAL = QANONHND * ITMCOST			
CALL REPORT VALUEREP			
END			
Required Name			

Figure 4-3 The VALUE Procedure Definition Window

Navigating from Definition Charts

Figure 4-4 shows a File Definition window. To move to the next lower window following this window, move the cursor down the chart to the segment that you want to select and press F8. Using Figure 4-4 as an example, selecting the segment CLASS opens the Segment Definition window for the CLASS segment shown in Figure 4-5.

Working:OLYMRSLT			
L=RELATION	ABOVE	ORPHANS	RETRIEVE DELETE X=EXPORT
File Name:OLYMRSLT	File Identifier:	Comments: No	FD0100
Record Format:UNDEFINED	Record Size:	Records Per Block:	Buffer Size:32760
<pre> graph TD CLASS[CLASS] --- EVEN[EVEN] EVEN --- MEDALS[MEDALS] </pre>			
Segment Name ; press F8 to define segment.			

Figure 4-4 Navigating from the File Definition Window

Navigating Upwards

The F3 function key is used to return to previous windows. When you press F3 on any window, you are always returned to the previous window (next higher) in the structure. For example, pressing F3 on the window shown in [Figure 4-5](#) opens the File Definition window shown in [Figure 4-4](#).

VISION:Workbench for DOS								Working:OLYMRSLT	
CENTER EDIT									
Segment Name.....:CLASS								FD0110	
Occurs N Times.....:									
Key Sequence.....:									
Field Name	Type	Loc	Len	Dec	Rnd	Key	Count	More Col	Attr Hdg
SPORTNM	CHARACTER	1	20			1			
SEG2CNT	FIXED POINT	21	4				EVENT		
Required Name									

Figure 4-5 Navigating Upwards

Function Keys

Function keys are available to obtain help information, to display names of items stored in the VISION:Workbench library, to exit windows and the system, and so on. When you press the appropriate function key, the respective action is taken.

There are two sets of function keys that can display at the bottom of a window. To view the function key options, press F1 or Alt + F1. Function keys display in context. That is, the function keys that display are the only permissible keys active within the context of the application.

Following is an explanation of each of the available function keys:

F1	HELP	Opens a Help window.
F2	NAMES	Displays a menu bar identifying the various objects or items available. Selecting an object or item displays a list of names.

F3	CLOSE	Closes the current window and positions the cursor within the previous window. If the cursor is on the menu bar, it returns to the current window. If the cursor is on the main menu, it returns to DOS.
F4	MENU BAR	Moves the cursor to the menu bar on the second line of the window.
F5	SAVE	Saves the object in memory to disk.
F6	CHOICES	Lists the available choices that can be made for specification entry.
F7	VALIDATE	Validates the object being displayed or the entire application depending upon the context.
F8	OPEN	Opens a window.
F9	INSERT	Inserts a line of text or an object.
Alt+F1	HELP	Displays the extended function key bar. To activate any of the extended function key capabilities, you need to press Alt in combination with the respective function key.
Alt+F3	ZOOM	A toggle switch that expands or contracts a window to or from a full size screen without closing any of the currently displayed windows.
Alt+F6	PRINT	Prints the object or application on the system printer.
Alt+F8	SWITCH	Makes a different window the active window. For detailed information, see the next section.
Alt+F10	DELETE	Deletes a line of text or an object.

Switch (Alt+F8)

Because more than one window can be open at a time, the SWITCH function key allows you to switch between windows, making a different window the active window (that is, the window where the cursor is). When SWITCH is activated, the cursor moves to the first field of the next open window. Each use of the SWITCH key moves from window to window in turn.

If the data displayed in a window is dependent upon the position of the cursor in another window, the data in the window changes as the cursor moves. The HELP information provides a good example of how SWITCH works. If, for example, you press HELP for the Field Name entry on the Field Definition window, the HELP for that entry displays. You can press Alt+F8 to make the field definition window active, and you can Tab from one field to another. As the cursor moves, the HELP window changes. While the cursor is in the Field Definition window, it is the active window and you can enter information.

As with all other window activities, you close a window by pressing F3. When you press F3, the cursor returns to the position that it occupied in the first window prior to the SWITCH that opened the additional window. That is, it is exactly as if the additional window had never been opened.

Button Fields

A button field is any displayable field that is a selection or summarization of an underlying window. Button fields normally contain a number (a count of underlying items) or YES/NO. When the cursor is positioned on a button field and you press the OPEN (F8) key, a follow-on window opens.

Entering Text

To enter information in VISION:Workbench, just type. For each character key you press, that character appears at the current cursor position, replacing whatever is there. Then, the cursor moves one position to the right.

To correct a mistake, use the arrow keys to move the cursor to the position you want to change; type the correct information. You can also use the Ins (insert) and Del (delete) keys to insert and delete information.

Entry Descriptions

The choice or choices that you have for each window entry are listed in the lower-left corner of the window. You can look at these and see the choices that are available whenever you are unsure about what to enter.

When you move your cursor to other window entries, the information in the lower-left corner changes to show the entries available for those entries. [Figure 4-6](#) shows the entry description for the PC Display Monitor entry on the Setup window.

VISION:Workbench for DOS										Current:	
Library Path :										Setup Specifications	US0100
Primary C:\MPCXX\SAMPAPPS											
Import Input Path:											
C:\MPCXX\IMPORT											
Export Output Path:											
C:\MPCXX\EXPORT											
Printer Output Dest:											
PRN											
Printer Setup: Yes										PC Display Monitor:	COLOR
Target Mainframe Support											
Native ASL: YES											
F1	F2	F3	F4	F5	F6	F7	F8	F9	Alt+F10		
Help		Close		Save							
CHOICES: <blank>, MONOCHROME, COLOR, OTHER											

Figure 4-6 Entry Descriptions

You can also press F6 (CHOICES) to display a list of choices including descriptions. You can scroll up and down through the list until the wanted entry is highlighted. Press Enter to choose the highlighted item, and the selection is placed in the field. Press F3 or Escape to exit the choice list without making a selection.

For additional information about an entry, you can request help information by pressing F1 twice.

Help Information

You can obtain descriptive help information for any window entry at any time. With your cursor positioned on the entry, press the help key (F1) twice: once to display the function key bar at the bottom of the window (as shown in [Figure 4-7](#)) and a second time to display the help information for that entry.

VISION:Workbench for DOS				Working:ITEM					
L=RELATION ABOVE ORPHANS RETRIEVE DELETE X=EXPORT									
File Name:ITEM		File Identifier:		Comments: No		FD0100			
Record Format: VARIABLE		Record Size:		Records Per Block:		Buffer Size:3120			
ITEMSEG									
F1	F2	F3	F4	F5	F6	F7	F8	F9	Alt+F10
Help	Names	Close	Menu Bar	Save	Choices	Validate	Open	Insert	Delete

Figure 4-7 The Function Key Bar

The second time you press F1, a window opens containing the help information (see [Figure 4-8](#)). The window is opened in a portion of the window that does not obscure the window entries. To see the help information for each entry while you continue to enter text, use the SWITCH (Alt+F8) function key. Then you can move your cursor to other window entries; help information for each window entry displays in the same window.

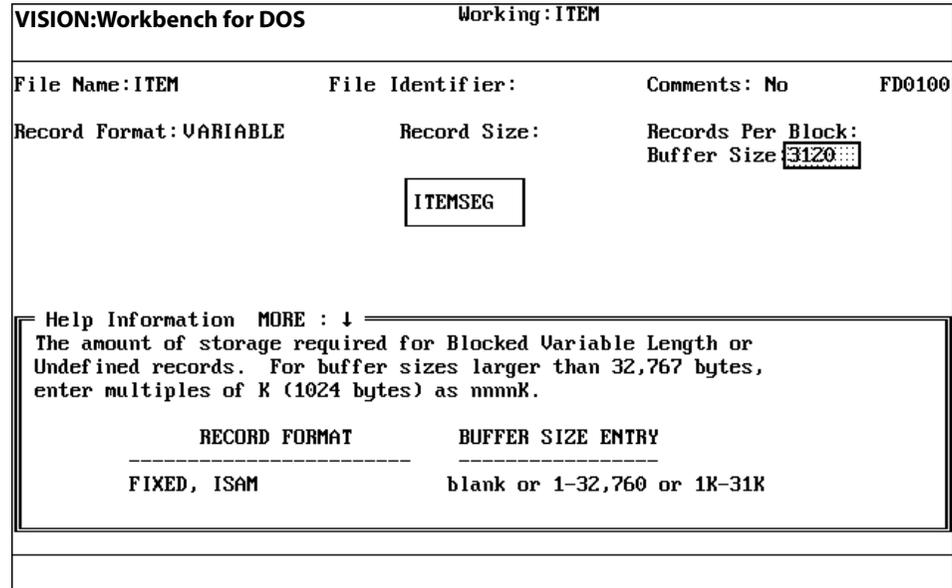


Figure 4-8 A Help Information Window

Workbench Prompts

VISION:Workbench uses the bottom line of the window to display prompt messages. These messages provide you with information about an operation VISION:Workbench is about to perform. They are also issued as the result of selecting certain menu items.

Prompts provide descriptive information and assist you in making a decision whether to proceed with the operation or not. You respond to the prompt by keying in the requested information. [Figure 4-9](#) shows an example of a commonly issued prompt.

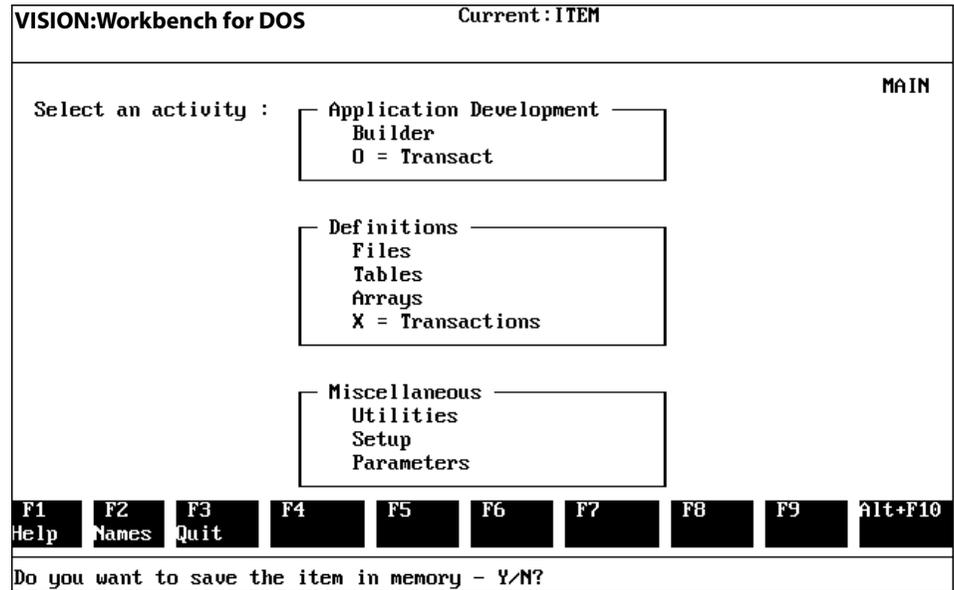


Figure 4-9 VISION:Workbench for DOS Prompt

The prompt shown in [Figure 4-9](#) is issued because an application component is being edited. You must specify whether to save it before allowing editing of another application or termination of VISION:Workbench.

Using the Editing Menu Bar Items

Menu bars are used to invoke specific functions related to the window currently displayed. Menu bars display on the second line of the window. You move the cursor to the menu bar by pressing the F4 key. The editing menu bar items are available to help manipulate new or existing application and definition text. You can use the COPY, MOVE, SEARCH, and REPLACE menu bar items while entering or editing applications and definitions.

Application windows that list report, procedure, and subfile objects display the editing menu bar items. [Figure 4-10](#) shows an example of a window with the menu bar displayed on the second line of the window.

VISION:Workbench for DOS		Working:EX1
COPY: MOVE SEARCH REPLACE		
Procedure Name:VALUE	Type : N	RQ0100
Pre-Selection : N	Re-init Temps:	Maximum Items Selected:
;EACH MASTER-FILE RECORD CONTROL PROCEDURE ; STOCKVAL: FIELD P 5 DEC 2 IF QANONHND > 0 LET T.STOCKVAL = QANONHND * ITMCOST CALL REPORT VALUEREP END		
Required Name		

Figure 4-10 The Editing Menu Bar

Press F4 to display the editing menu bar. Select one of the editing menu bar items:

- Move the cursor to it and press Enter.
- Enter its first letter (for example, entering a C selects COPY).

The COPY menu bar item is used to copy one or more complete lines of text from one place on the window to another. Follow the prompts appearing at the bottom of the window to copy the text.

The MOVE menu bar item is used to move one or more completed lines of text from one place on the window to another. Once the MOVE operation is completed, the text no longer exists in its previous location. Follow the directions appearing at the bottom of the window to MOVE the text.

While the functions of COPY and MOVE are the same for screen painting, they work with blocks of data rather than lines of data. This is explained in [Chapter 10. Screen Layout, Contents, and Actions](#).

The SEARCH menu bar item is used to search for strings of text. Follow the directions appearing at the bottom of the window to search for the text.

The REPLACE menu bar item is used to search for a string and replace it with the replacement string that you specify. Any string previously searched for displays in the search string prompt message when REPLACE is invoked. Follow the directions appearing at the bottom of the window to search for and replace the text when found.

In the definition component, the COPY, MOVE, SEARCH, and REPLACE menu bar items are accessed a little differently. The CENTER and EDIT menu bar items are shown when the window is first displayed (as shown in [Figure 4-11](#)). You must select EDIT to display the editing menu bar (shown in [Figure 4-10](#)).

The screenshot shows the VISION:Workbench for DOS interface. At the top, the title bar reads "VISION:Workbench for DOS" on the left, "Working:ITEM" in the center, and "Ver: 6.00" on the right. Below the title bar, a menu bar contains "CENTER" and "EDIT". The main window displays the following information:

```

Segment Name.....:ITEMSEG
Occurs N Times.....:
Key Sequence.....:
  
```

Field Name	Type	Loc	Len	Dec	Rnd	Key	Count	More Attr	Col Hdg
ITEMNO	CHARACTER	1	7			1			Y
ITEMNAME	CHARACTER	8	20						Y
ITEMPRICE	ZONED	28	6	2				Y	Y
ITEMCOST	ZONED	34	6	2				Y	Y
ITEMWGT	ZONED	40	3						Y
QANONHND	ZONED	43	5						Y
SHIPYTD	ZONED	48	6						Y
ORDQTY	ZONED	54	6						Y
QTYBKORD	ZONED	60	5						Y
REORDPT	ZONED	65	4						Y
REORDQTY	ZONED	69	4						Y
PRODCODE	CHARACTER	73	2						Y
VENDOR	CHARACTER	75	4						Y
FILLER	FIXED POINT	79	1						

Required Name

Figure 4-11 The CENTER and EDIT Menu Bar

Documenting Applications

VISION:Workbench allows you to document as much of your application as you like. When you select the COMMENTS button field on the Application window, a Comments window displays. You can enter as much descriptive information on this window as you like.

You can page down to enter more lines of text when the bottom of the window is reached. Comments are stored along with your application, and you can change them in future sessions if the need arises. A similar Comments window can also be stored with file, table, array, and transaction definitions. [Figure 4-12](#) shows a documented application.

VISION:Workbench for DOS	Working:EX1
COPY MOVE SEARCH REPLACE	
Comments for : Builder EX1	CM0100
Sample application	
1. The application is designed as a walk-through.	
2. It uses the ITEM file.	
3. It creates a report.	
Enter a comment line	

Figure 4-12 Documenting Applications

Modifying the Application

You can modify any portion of an existing application after it is created (for example, definitions, reports, screens, screen flow, and procedures).

Adding Text Lines

You add text to any application object by pressing the down arrow key until the cursor is positioned on the first blank line following the last text line. On the blank line, you can enter the text to be added. See [Figure 4-13](#).

The screenshot shows the 'CENTER EDIT' screen in UISION Workbench for DOS. The title bar indicates 'Working: ITEM' and 'Ver: 6.00'. The main area displays the following information:

```

Segment Name.....:ITEMSEG
Occurs N Times....:
Key Sequence.....:
  
```

Field Name	Type	Loc	Len	Dec	Rnd	Key	Count	More Attr	Col Hdg
ITEMNAME	CHARACTER	8	20					Y	Y
ITMPRIE	ZONED	28	6	2				Y	Y
ITMCOST	ZONED	34	6	2				Y	Y
ITEMMGT	ZONED	40	3						Y
QANONHND	ZONED	43	5						Y
SHIPYTD	ZONED	48	6						Y
ORDQTY	ZONED	54	6						Y
QTYBKORD	ZONED	60	5						Y
REORDPT	ZONED	65	4						Y
REORDQTY	ZONED	69	4						Y
PRODCODE	CHARACTER	73	2						Y
VENDOR	CHARACTER	75	4						Y
FILLER	FIXED POINT	79	1						Y

At the bottom of the screen, the text 'Name of Field' is visible, indicating the current cursor position.

Figure 4-13 Adding a Text Line

Inserting Text Lines

You insert lines of text into an existing application component by pressing F9. Position the cursor on the text line that you want to insert the new text immediately in front of and press F9. A blank line opens up at that location, and the cursor is positioned at the leftmost position on the line. You can begin entering text at that location. See [Figure 4-14](#).

Field Name	Type	Loc	Len	Dec	Rnd	Key	Count	More Attr	Col Hdg
ITEMNO	CHARACTER	1	7			1			Y
ITEMNAME	CHARACTER	8	20						Y
ITEMPRICE	ZONED	28	6	2				Y	Y
ITEMCOST	ZONED	34	6	2				Y	Y
ITEMMGT	ZONED	40	3						Y
QANONMIND	ZONED	43	5						Y
SHIPYTD	ZONED	48	6						Y
ORDQTY	ZONED	54	6						Y
QTYBKORD	ZONED	60	5						Y
REORDPT	ZONED	65	4						Y
REORDQTY	ZONED	69	4						Y
PRODCODE	CHARACTER	73	2						Y
VENDOR	CHARACTER	75	4						Y

Figure 4-14 Inserting a Text Line

If you cannot remember which function key performs the insert, press F1 to get the function key bar which will remind you about F9.

Deleting Text Lines

You delete lines of text by using the Alt key and the F10 key.

Position the cursor on the line that you want to delete.

Press the Alt key and the F10 key. The line is deleted, and the remaining lines move up.

If errors are detected during validation, you are notified of their presence and given the opportunity to correct them. [Figure 4-16](#) shows a window with validation errors.

Field Name	Type	Loc	Len	Dec	Rnd	Key	Count	More Attr	Col Hdg
ITEMNO	CHARACTER	1	7			1		y	y
ITEMNAME	CHARACTER	8	20					y	y
ITEMPRICE	ZONED	28	6	2				y	y
ITEMPCOST	ZONED	34	6	2				y	y
ITEMWGT	ZONED	40	3					y	y
QANOMIND	ZONED	43	5					y	y
SHIPYD	ZONED	48	6					y	y
ORDQTY	ZONED	54	6					y	y
QTYBKORD	ZONED	60	5					y	y
REORDPT	ZONED	65	4					y	y
REORDQTY	ZONED	69	4					y	y
PRODCODE	CHARACTER	73	2					y	y
HENDOR	CHARACTER	75	4					y	y
FILLER	FIXED POINT	79	1						

Required Name - all blanks

Figure 4-16 Validation Errors

Saving Applications

There are two ways to save application objects and entire applications with VISION:Workbench:

- Press the F5 key at any time during your application development to save a copy of the application object on which you are working.
- When you enter another component of VISION:Workbench (for example, going from Definitions to Builder), the following message displays:

"Do you want to save the item in memory - Y/N?"

Press Y to save a copy of the application. Press N to not save a copy of the application; any changes that you made after the last SAVE (F5) are completely lost.

Deleting Applications and Application Objects

You can delete entire applications, individual objects of applications, or definitions from the VISION:Workbench library. You do this in the following way:

- Position the cursor on the item to be deleted.
- Hold down the Alt key and press the F10 key. VISION:Workbench displays the following prompt:

"Delete this item from application - Y/N?"

if you are deleting an application object or

"Delete xxx named yyy from library - Y/N?"

where xxx is the name of the component (for example, Builder, File, Table) and yyy is the name of the object to be deleted.

- Press Y to confirm that you want to proceed with the delete. Press N to abort the delete operation.

If you do not remember what function keys to press for a delete, you can press the F1 key for help. The function key bar displays at the bottom of the window listing all the available function keys and their usage.

It is a good idea to create periodic backup copies of your application directory. In the event that you inadvertently delete a part of your application, you can easily restore it from your backup.

Definition Component

This chapter describes the use of the definition component of VISION:Workbench. The definition component is used to define files, tables, arrays, and transactions used by your application. You enter it by selecting any of the Main Menu items under the Definitions heading. [Figure 5-1](#) shows the structure of how the windows within the definition component are organized.

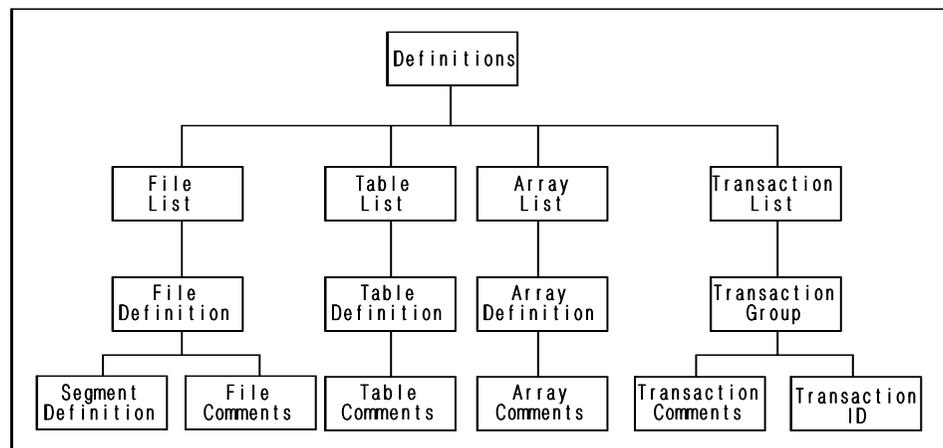


Figure 5-1 Definition Component Structure

The file definition windows are used for both the VISION:Transact and VISION:Builder components of the system. In fact, the same definition can be used by either a VISION:Transact or VISION:Builder application. Tables, arrays, and transactions can only be used in VISION:Builder applications.

Definition Lists

In [Figure 5-1](#), note that each type of definition has a list window associated with it. These windows are the first to open when you select the definition type on the Main Menu. [Figure 5-2](#) shows a File List window.

Each list contains the definition type, file name, and directory location. If the file is currently in memory, this is indicated as the location instead of a directory name.

You can enter new names into the list at any time. Adding the name to the list opens the appropriate definition window for that type.

VISION:Workbench for DOS		Current :
USERINFO	User file index information	FI0100
Type	Name	Location
FILE	CUSTOMER	C:\MPCXX\SAMPAPPS
FILE	CUSTOMER	C:\MPCXX\SAMPAPPS
FILE	DEPTFILE	C:\MPCXX\SAMPAPPS
FILE	ITEM	C:\MPCXX\SAMPAPPS
FILE	ITEMASTX	C:\MPCXX\SAMPAPPS
FILE	ITEMX	C:\MPCXX\SAMPAPPS
FILE	OLYMRSLT	C:\MPCXX\SAMPAPPS
FILE	PERSFILE	C:\MPCXX\SAMPAPPS
FILE	TRANWORK	C:\MPCXX\SAMPAPPS
FILE	WORKSTOR	C:\MPCXX\SAMPAPPS
Name of item: press F8 to select this item.		

Figure 5-2 Definition List Window

File Definitions

Files are used in both VISION:Transact and VISION:Builder applications. They are defined using four windows:

- File List.
- File Definition.
- File Comments (optional).
- Segment Definition.

File List Window

Lists all existing file definitions and can be used to enter new definition names (see [Figure 5-2](#)).

File Definition Window

Specifies the file characteristics and file structure. The file structure displays graphically on the window. It represents the file as a hierarchical tree of segments. The structure illustrates the parent-dependent relationships of the segments.

You can insert and/or delete boxes in the structure, creating or deleting segments. [Figure 5-3](#) shows an example of a File Definition window.

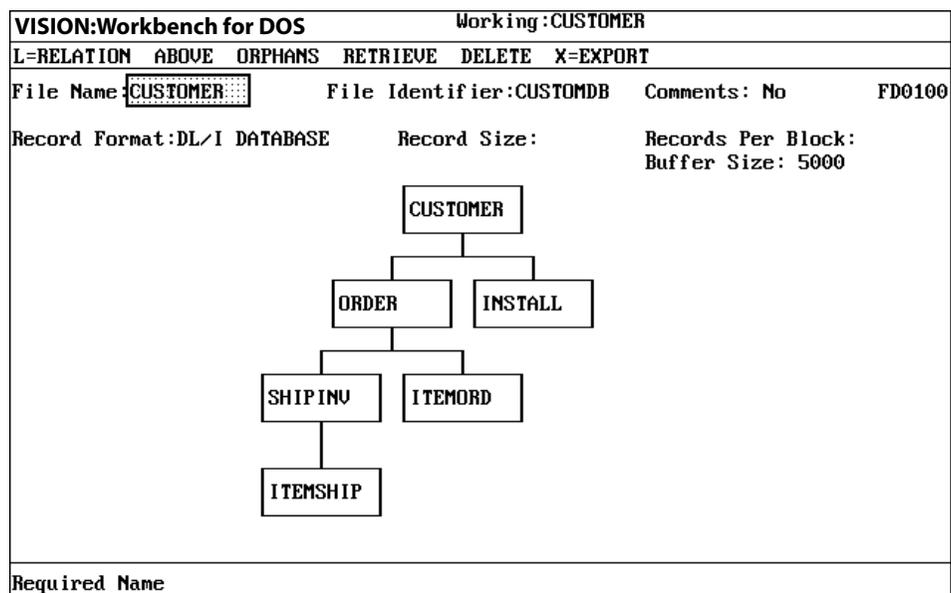


Figure 5-3 File Definition Window

File Comments Window

Allows you to enter comments or descriptive information about the file. To open the window, place the cursor over the Comments button field and press F8. The comments you enter are preserved through the import and export facilities.

Segment Definition Window

Specifies the segment characteristics, fields within the segment, column headings to be used in subsequent reports, and output editing information for the fields. There are some field-level specifications that are not used by the VISION:Transact (for example, field editing characteristics and column headings). Completing these specifications does not impact the VISION:Transact application; the information is not used.

Figure 5-4 shows a Segment Definition window. (See [Defining Files on page 5-9](#) for more detailed information about defining files.)

Field Name	Type	Loc	Len	Dec	Rnd	Key	Count	More Attr	Col Hdg
ITEMNO	CHARACTER	1	7			1			Y
ITEMNAME	CHARACTER	8	20						Y
ITEMPRICE	ZONED	28	6	2				Y	Y
ITEMCOST	ZONED	34	6	2				Y	Y
ITEMWGT	ZONED	40	3						Y
QANONHND	ZONED	43	5						Y
SHIPYTD	ZONED	48	6						Y
ORDQTY	ZONED	54	6						Y
QTYBKORD	ZONED	60	5						Y
REORDPT	ZONED	65	4						Y
REORDQTY	ZONED	69	4						Y
PRODCODE	CHARACTER	73	2						Y
UENDOR	CHARACTER	75	4						Y
FILLER	FIXED POINT	79	1						

Figure 5-4 Segment Definition Window

Table Definitions

Table definitions are used in VISION:Builder applications only. They are defined using three windows:

- Table List.
- Table Definition.
- Table Comments (optional).

Table List Window

Displays a list of all existing table definitions and can be used to create new table definition names (similar to [Figure 5-2](#)).

Table Definition Window

Specifies the table contents, table type, valid arguments, argument characteristics, and argument results. All valid entries display in the lower-left corner of the window. For further details of each entry, refer to the help information. [Figure 5-5](#) shows a Table Definition window.

VISION:Workbench for DOS		Working:NUMCNTY	
EDIT X=EXPORT			
Table Name	NUMCNTY	Comments: No	FD0300
Table Type...	BINARY		
Argument Type	ZONED	Length: 2	Decimals:
Result Type	CHARACTER	Length: 15	Decimals:
Argument	Result		
1	AUSTRALIA		
2	AUSTRIA		
3	CANADA		
4	DENMARK		
5	FRANCE		
6	EAST GERMANY		
7	WEST GERMANY		
8	GREECE		
9	HOLLAND		
10	ITALY		
11	JAPAN		
12	MEXICO		
13	NEW ZEALAND		
Required Name			

Figure 5-5 Table Definition Window

Table Comments Window

Allows you to enter comments or descriptive information about the table. To open the window, place the cursor over the Comments button field and press F8. The comments you enter are preserved through the import and export facilities.

Array Definitions

Arrays are used in VISION:Builder applications only. They are defined using three windows:

- Array List.
- Array Definition.
- Array Comments (optional).

Array List Window

Displays a list of all existing array definitions and can be used to create new array definition names (similar to [Figure 5-2](#)).

Array Definition Window

Specifies the array characteristics, such as the number of rows and columns in the array. It also defines the array field entries, their location, length, and size. Column headings for subsequent reports and automatic table lookup arguments and results can also be specified. All valid entries display in the lower-left corner of the window. For further details of each entry, refer to the help information. [Figure 5-6](#) shows an Array Definition window.

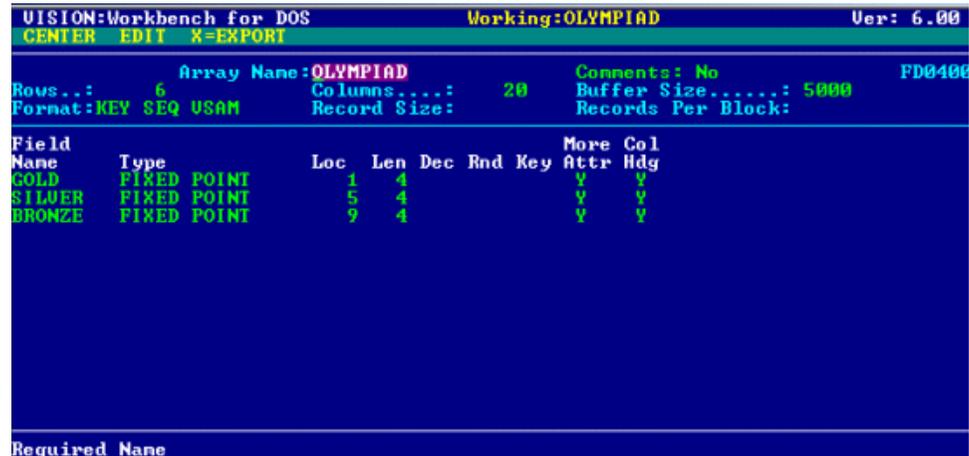


Figure 5-6 Array Definition Window

Array Comments Window

Allows you to enter comments or descriptive information about the array. To open the window, place the cursor over the Comments button field and press F8. The comments you enter are preserved through the import and export facilities.

Transaction Definitions

Transactions are used in VISION:Builder applications only. Transaction definitions are defined using four windows:

- Transaction List.
- Transaction Group.
- Transaction Definition (one or more).
- Transaction Comments (optional).

Transaction File List Window

Displays a list of all existing transactions or transaction groups and is used to enter new transaction definition names (similar to [Figure 5-2](#)).

Transaction Group Window

Identifies the file containing the transaction records, identifies the master file to be updated, and names the transactions contained in the group. [Figure 5-7](#) shows a Transaction Group window.

VISION:Workbench for DOS		Working:ORDENTRY											
X=EXPORT RETRIEVE													
Transaction Group: ORDENTRY		Master File Name: CUSTMAST	FD0200										
Comments.....: No		Transaction File: TRANWORK											
<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>ADDORDR</td> <td>ADD ORDER TRANSACTION</td> </tr> <tr> <td>ADDITEM</td> <td>ADD ITEM TO ORDER TRANSACTION</td> </tr> <tr> <td>DELORDR</td> <td>DEL ORDER TRANSACTION</td> </tr> <tr> <td>DELITEM</td> <td>DEL ITEM FROM ORDER TRANSACTION</td> </tr> </tbody> </table>		Name	Description	ADDORDR	ADD ORDER TRANSACTION	ADDITEM	ADD ITEM TO ORDER TRANSACTION	DELORDR	DEL ORDER TRANSACTION	DELITEM	DEL ITEM FROM ORDER TRANSACTION		
Name	Description												
ADDORDR	ADD ORDER TRANSACTION												
ADDITEM	ADD ITEM TO ORDER TRANSACTION												
DELORDR	DEL ORDER TRANSACTION												
DELITEM	DEL ITEM FROM ORDER TRANSACTION												
Required Group Name													

Figure 5-7 Transaction Group Window

Transaction Definition Window

One Transaction Definition window must be completed for each transaction named in the Transaction Group window. This window specifies the transaction identifier, the fields to be updated in the master file, and the actions to be taken against them. It also specifies the validation criteria to be performed against those fields.

Figure 5-8 shows a Transaction Definition window. (See [Defining VISION:Builder Transactions on page 5-18](#) later in this chapter for more detailed information about defining transactions.)

VISION:Workbench for DOS			Working:ORDENTRY	
COPY MOVE SEARCH REPLACE				
			Transaction Group:ORDENTRY	FD0250
			Master File.....:CUSTMAST	
			Transaction File.:TRANWORK	
ID name	ADD.ITEM	Description:ADD ITEM TO ORDER TRANSACTION		
		Name	Oper	Code
Transaction Identifier No. 1 :		TRANID1	EQ	4ADD
AND Transaction Identifier No. 2 :		TRANID2	EQ	ITEM
Transaction Field	Action Code	Master Field	Validation Type	Validation Criteria
CUSTNO	MATCH WITH	CUSTNO	PATTERN	Y9999
ORDERNO	MATCH WITH	ORDERNO		
ITEMORD	UPDATE/INSERT	ITEMORD		
ITEMNAME	XFER DATA ONLY	ITEMNAME	MIN/MAX	0.00 300.0
ITMPRICE	XFER DATA ONLY	ITMPRICE		
ITMQTYOR	XFER DATA ONLY	ITMQTYOR		
Required name				

Figure 5-8 Transaction Definition Window

Transaction Comments Window

Allows you to enter comments or descriptive information about the transaction. To open the window, place the cursor over the Comments button field and press F8. The comments you enter are preserved through the import and export facilities.

Defining Files

Files must be defined for both VISION:Transact and VISION:Builder applications. The File Definition window has many important entries. Being aware of these entries and their relationship to a successful validation of your application can save you time and needless frustration. Refer to the help information for details of each entry.

Using the Segment Structure Chart

The segment structure chart displays on the File Definition window. This window opens immediately following the file definition list for all new or existing files. You define the file structure by naming the segments of the file in rectangular boxes arranged in a tree-like structure.

The top portion of the window is used to specify the characteristics of the file (file identifier, record format, record size, records per block, and buffer size). The segment structure chart displays below these two lines. [Figure 5-9](#) shows a fully defined segment structure. Each box in the structure represents a segment. The name inside the box is the segment name. The box immediately above it is the parent; the box immediately below it is the dependent.

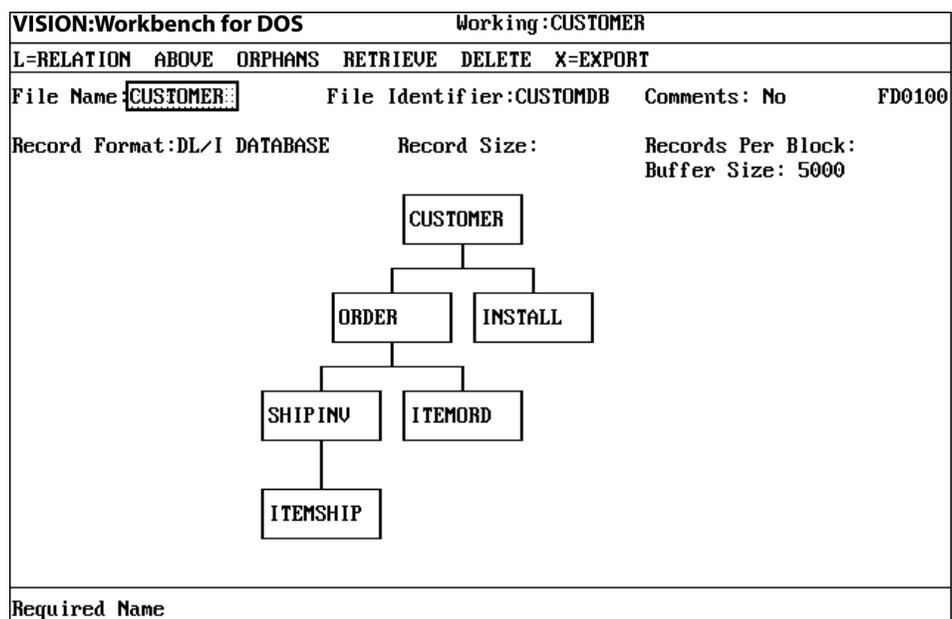


Figure 5-9 Segment Structure Chart

Navigating Through the Segment Chart

The arrow keys (up, down, left, and right) and the tab keys (shifted and unshifted) are used to navigate through the segment structure chart. The following describes the keys to use and their resultant cursor actions:

- Down arrow moves the cursor down one level; if a box does not exist, one is created.
- Up arrow moves the cursor up one level (except at the root level – the cursor moves to the file characteristics).
- Right arrow moves the cursor to the right one character position at a time until it reaches the end of the box; then it moves to the next box to the right if one exists. If a box does not exist, one is created.
- Left arrow moves the cursor left one character position at a time, until it reaches the end of the box; then it moves to the next box. If a box does not exist, one is created.
- Tab moves the cursor to the next box to the right. If a box does not exist, one is created. Pressing Tab on a blank box moves the cursor up one level, and the blank box disappears.
- Tab shifted moves the cursor to the next box to the left. If a box does not exist, one is created. Pressing Tab (shifted) on a blank box moves the cursor up one level, and the blank box disappears.

Defining a New File

You use the same navigation keys to create a new file definition that you use for navigating through an existing segment structure.

The first time that you enter the File Definition window, the file name entry at the top of the window contains the name you assigned it on the definition list. A single, empty, rectangular box displays below the characteristic portion of the window. This is for naming the first (root) segment in the file. [Figure 5-10](#) shows an example of what this looks like.

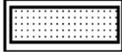
VISION:Workbench for DOS		Working:NEWFILE	
L=RELATION ABOVE ORPHANS RETRIEVE DELETE X=EXPORT			
File Name:NEWFILE	File Identifier:	Comments: No	FD0100
Record Format:	Record Size:	Records Per Block:	Buffer Size:
			
Required Name			

Figure 5-10 Segment Structure Window for a New Definition

To complete this window definition

1. Enter the file characteristics information on the top portion of the window, and the cursor moves to inside the empty box (the root segment).
2. Enter a name in the box, and you have created the first segment.
3. Continue to define the other segments in the structure or press F8 to define the fields in this first segment. (Pressing F8 brings up the Segment Definition window.)
4. Press the Tab key (or down arrow) to continue defining the next level of the structure. This moves you to the next level, below the first box. A new box displays underneath the first box.
5. Enter the next segment name.
6. Use the Tab key to define new segments on the same level (level two).

[Figure 5-11](#) shows a new segment to be defined on the right (by pressing Tab). [Figure 5-12](#) shows a new segment to be defined on the left (by pressing Shift Tab).

Define new dependent segments at the next level by pressing the down arrow. A new empty box with the cursor positioned inside appears directly below the box on which you pressed the down arrow. (See [Figure 5-13](#).)

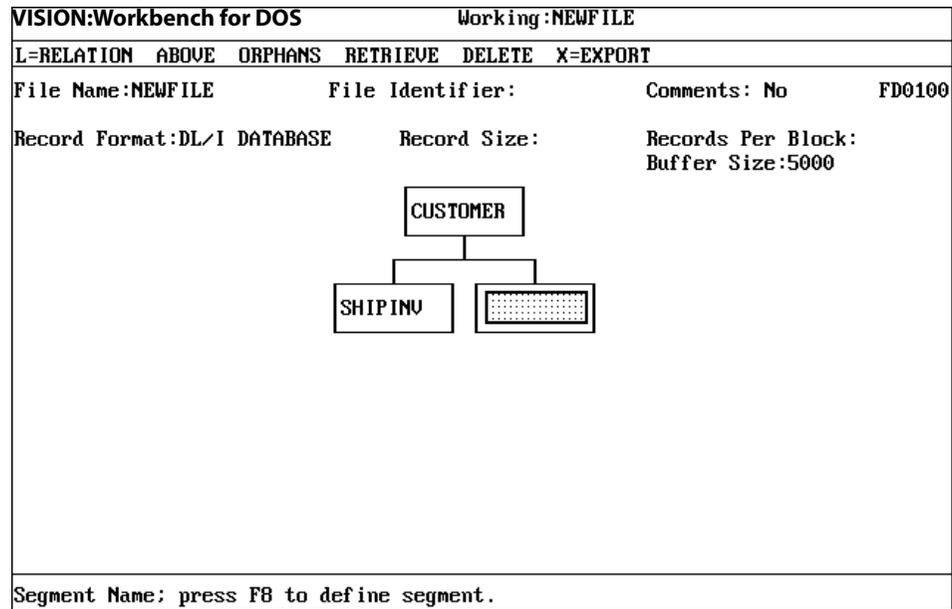


Figure 5-11 A New Box on the Same Level on the Right

VISION:Workbench for DOS		Working:NEWFILE	
L=RELATION ABOVE ORPHANS RETRIEVE DELETE X=EXPORT			
File Name:NEWFILE	File Identifier:	Comments: No	FD0100
Record Format:DL/I DATABASE	Record Size:	Records Per Block:	Buffer Size:5000
<pre> graph TD CUSTOMER[CUSTOMER] --- H[] H --- Dotted[] H --- SHIPINV[SHIP INV] </pre>			
Segment Name; press F8 to define segment.			

Figure 5-12 A New Box on the Same Level on the Left

VISION:Workbench for DOS		Working:NEWFILE	
L=RELATION ABOVE ORPHANS RETRIEVE DELETE X=EXPORT			
File Name:NEWFILE	File Identifier:	Comments: No	FD0100
Record Format:DL/I DATABASE	Record Size:	Records Per Block:	Buffer Size:5000
<pre> graph TD CUSTOMER[CUSTOMER] --- SHIPINV[SHIP INV] SHIPINV --- Dotted[] </pre>			
Segment Name; press F8 to define segment.			

Figure 5-13 A New Box on a New Level

7. At any time or at the conclusion of the structure definition, you can move back up the structure by using the up arrow. Pressing the up arrow from a newly opened box causes the box to disappear and the cursor to move up one level.
8. Position the cursor on each segment to define the segment's fields and press F8. You can define fields for segments as you name them in the structure or at any point in this process.

Deleting Segments from the Structure

There are two ways to remove segments from the structure chart. If you use the DELETE menu bar item (discussed in the next section), the segment is completely erased from the system. If you use the method described below, the segment is removed from the chart but is placed in the orphan list.

1. Position the cursor in the segment box to be deleted.
2. Press the Alt+F10 keys. The box is removed from the chart, and the segment name is placed in the orphan list. (Orphans are discussed later in this chapter.)

Definition Menu Bar Items

The file definition component menu bars display on the File Definition window and Segment Definition window. The menu bar items display on the second line of the window, as shown in [Figure 5-14](#). To move the cursor to the menu bar, press F4. You select the item that you want by typing its first letter or by moving the cursor to it and pressing Enter. The menu bar items are available during the definition process and are discussed later in this chapter.

The following apply to the File Definition window:

- **RELATION** – allows you to enter a logical expression that defines the relationship of the table corresponding to this segment to the table corresponding to the parent segment.
- **ABOVE** – in conjunction with the F9 insert key, inserts a blank segment box above any box from which this item is selected.
- **ORPHANS** – provides a list of orphan segments that are available in memory but are unattached to any application. Once an orphan is selected for use in an application, it is removed from the orphan list.
- **RETRIEVE** – retrieves segments from VISION:Workbench library into the orphan list and, for transaction definitions, includes orphan transactions for transaction groups.
- **DELETE** – provides a list of segment names. Once a segment is selected for deletion, it is erased from the system.
- **EXPORT** – generates a copy of the definition for subsequent submission to the host.

The following two items apply to the Segment Definition window:

- **EDIT** – displays the COPY, MOVE, SEARCH, REPLACE menu bar on the Segment Definition window.
- **CENTER** – is a toggle that is used to specify whether report column headings are to be centered in printed reports.

VISION:Workbench for DOS		Working:NEWFILE	
L=RELATION ABOVE ORPHANS RETRIEVE DELETE X=EXPORT			
File Name:NEWFILE	File Identifier:CUSTOMDB	Comments: No	FD0100
Record Format:DL/I DATABASE	Record Size:	Records Per Block: Buffer Size: 5000	
<div style="display: flex; flex-direction: column; align-items: center; gap: 10px;"> <div style="border: 1px solid black; padding: 2px 10px;">CUSTOMER</div> <div style="border: 1px solid black; padding: 2px 10px;">SHIPINU</div> <div style="border: 1px solid black; padding: 2px 10px; width: 50px; height: 20px;"></div> </div>			
Select box name from segment list			

Figure 5-14 Definition Component Menu Bar

Using RELATION

The RELATION menu item allows you to enter a logical expression that defines the relationship between the DB2® table corresponding to this segment and its parent. The logical expression can use any of the relational operators EQ, NE, LE, GE, LT, GT, and IS and the logical operators AND, OR, and NOT. The left operand of a relational operator can be any DB2 column name for the table. The right operand of a relational operator must be a field name defined within this file definition in any parent segment or the reserved words NULL or NOT_NULL in the case of the IS operator. You can use parentheses to override the order of evaluation of the expression. For example:

```
ORDER_CUSTNUM EQ CUSTNUM
(A EQ B AND C IS NULL) OR (D NE E AND F IS NOT_NULL)
```

Using ABOVE

The ABOVE menu item allows you to insert a new segment above an existing segment. You select it on the File Definition window. You use ABOVE by positioning your cursor in a segment box that you want to insert a new segment immediately above. After selecting ABOVE, press the F9 insert key.

A new blank box displays immediately above the box in which your cursor was positioned (see [Figure 5-15](#)); the cursor is repositioned in the new box. You can type a segment name into the new box, creating a new segment.

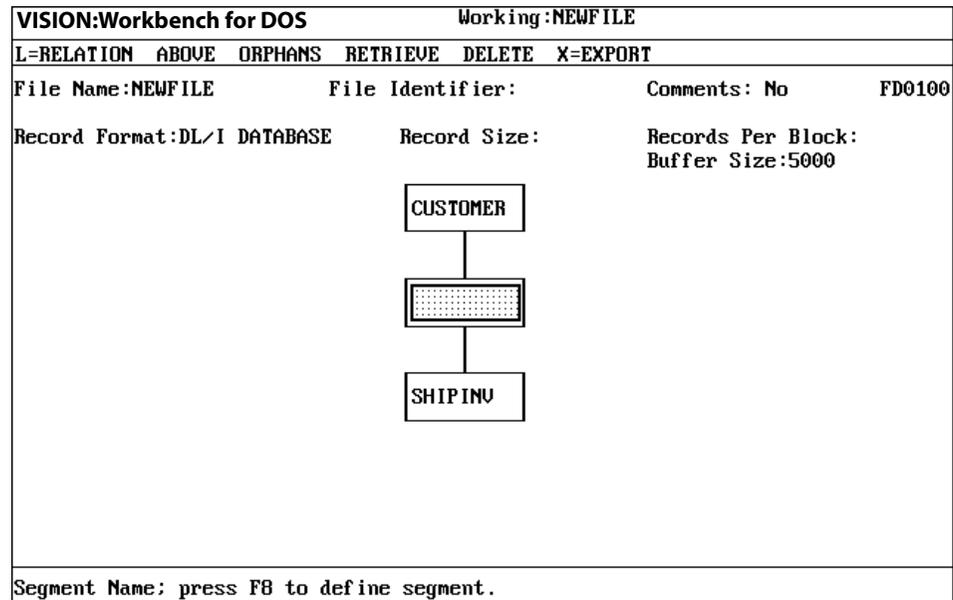


Figure 5-15 New Segment Box After Selecting ABOVE

Using ORPHANS

The ORPHANS menu item displays a list of segments available for use in your structure. These are segments that have been previously deleted using Alt+F10 function key. Orphans are segments or items that exist but are unattached to an application. Once an orphan is used in an application, it is removed from the list.

To use an orphan, you must first create an empty box in the structure where you want the orphan inserted. Press F4 to move the cursor to the menu bar. Select ORPHANS from the menu bar and position the cursor over the orphan you want to select. Press Enter to remove the orphan from the orphan list and place it into the structure.

Using RETRIEVE

The RETRIEVE menu item allows you to retrieve items from the library. VISION:Workbench allows you to save individual segments of a definition into a library for later use.

To use these items that have been saved, they must first be retrieved into memory. Using RETRIEVE allows you to read the selected items from the library into memory, making them available for other definitions. These retrieved items become ORPHANS and are placed on the orphan list. (See [Using ORPHANS on page 5-17.](#))

Using DELETE

DELETE displays a list of segments that are currently in your file structure. When you select a segment, it is deleted from the structure and the system (that is, it is not placed in the orphan list).

Using EXPORT

EXPORT generates a definition to be subsequently submitted to the host. If the definition has not been previously validated, requesting export will display a message that asks if you want to suppress or select validation of the definition. If validation is suppressed, the application may be exported with errors.

Using EDIT

When you select EDIT on the Segment Definition window, the COPY, MOVE, SEARCH, REPLACE menu bar displays. (See [Using the Editing Menu Bar Items on page 4-11](#) for detailed information about the editing menu bar items.)

Using CENTER

The CENTER menu bar item works as a toggle on the Segment Definition window. It determines whether column headings are to be centered. Centered column headings are the default. The toggle works in the following way:

- Toggle ON – column headings are centered.
- Toggle OFF – column headings are not centered.

You can tell whether it is on or off by its brightness. The CENTER menu bar item is highlighted when centering is turned on; it is dim when it is off. (Move the cursor from CENTER to EDIT to see if it is bright or dim.)

You turn off centering by selecting CENTER. When centering is off, you can position column headings wherever you want. The position you place the heading in determines its position within the report column.

To center an existing column heading

1. Turn on centering if it is not already on.
2. Type over at least one position in each line of the column heading.
3. Press Enter. The heading is automatically centered.

Defining VISION:Builder Transactions

Transactions are used in VISION:Builder applications only. Before you define a transaction group using VISION:Workbench, you must have a file definition of the transaction record layout, as well as a file definition of the master file to be updated. Both of these file definitions must be validated and stored in VISION:Workbench library before you can successfully validate the transaction group.

Transaction groups describe updating operations to be performed on the master file. Transaction groups specify the following:

- The master file to be updated.
- The transaction file containing the records to be applied to the master file.
- Actions to be taken against master file records.
- Transaction data input validation criteria.

Transaction groups can consist of one or more transactions. Each transaction specifies one or two codes, and 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 on the Transaction Definition window.

Identification of the transaction includes:

- 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.

The actions to be performed by the transaction can include:

- The transaction field to be used in the update.
- The segment and field actions to be taken against the master file field.
- The name of the master file field to be updated by the transaction field.
- Specifications for validating transaction data.

Defining the Transaction Group

A transaction list similar to the definition list shown in [Figure 5-2](#) displays after you select transaction definitions on the Main Menu. On the list window, you can enter the name of a new transaction group or you can select a transaction group from the list to be modified.

The Transaction Group window shown in [Figure 5-16](#) opens after you enter the transaction group name and press F8. On this window, you must enter the name of the master file to be updated and name of the file definition of the transaction file containing the records to be used in the update.

VISION:Workbench for DOS		Working: TRNGROUP	
X=EXPORT RETRIEVE			
Transaction Group:TRNGROUP		Master File Name:CUSTMAST	FD0200
Comments.....: No		Transaction File:CUSTTRAN	
Name	Description		
NEWCUST	ADD NEW CUSTOMERS TO MASTER FILE		
NEWORDER	ADD NEW ORDERS TO MASTER FILE		
NEWITEM	ADD NEW ITEMS TO MASTER FILE		
Required Group Name			

Figure 5-16 Defining a Transaction Group

In the example, TRNGROUP is specified as the transaction group name. CUSTMAST is the master file definition name, and CUSTTRAN is the name of the file definition that defines the transaction file records. You can define one or more transaction ID names for a transaction group. There are three transaction ID names in this group: NEWCUST, NEWORDER, and NEWITEM. To specify a transaction ID name, enter any valid 1- to 8- character name in the Name entry and press F8. This opens the Transaction Definition window, shown in [Figure 5-17](#).

Defining the Transaction

[Figure 5-17](#) shows an example of the completed NEWORDER transaction. Entries made on previous windows are carried forward to this window.

VISION:Workbench for DOS		Working: TRNGROUP																										
COPY		MOVE SEARCH REPLACE																										
Transaction Group:TRNGROUP		FD0250																										
Master File.....:CUSTMAST																												
Transaction File.:CUSTTRAN																												
ID name	<input type="text" value="NEWORDER"/>	Description:ADD NEW ORDERS TO MASTER FILE																										
Transaction Identifier No. 1 :	ID1	Name	Oper Code																									
AND Transaction Identifier No. 2 :	ID2		EQ ADD																									
			EQ ORDR																									
<table border="1"> <thead> <tr> <th>Transaction Field</th> <th>Action Code</th> <th>Master Field</th> <th>Validation Type</th> <th>Validation Criteria</th> </tr> </thead> <tbody> <tr> <td>CUSTNO</td> <td>UPDATE/INSERT</td> <td>CUSTNO</td> <td></td> <td></td> </tr> <tr> <td>ORDERNO</td> <td>INSERT</td> <td>ORDERNO</td> <td></td> <td></td> </tr> <tr> <td>ORDRDATE</td> <td>REPLACES</td> <td>ORDRDATE</td> <td></td> <td></td> </tr> <tr> <td>ORPERSON</td> <td>REPLACES</td> <td>ORPERSON</td> <td>PATTERN</td> <td>-9-9-9</td> </tr> </tbody> </table>		Transaction Field	Action Code	Master Field	Validation Type	Validation Criteria	CUSTNO	UPDATE/INSERT	CUSTNO			ORDERNO	INSERT	ORDERNO			ORDRDATE	REPLACES	ORDRDATE			ORPERSON	REPLACES	ORPERSON	PATTERN	-9-9-9		
Transaction Field	Action Code	Master Field	Validation Type	Validation Criteria																								
CUSTNO	UPDATE/INSERT	CUSTNO																										
ORDERNO	INSERT	ORDERNO																										
ORDRDATE	REPLACES	ORDRDATE																										
ORPERSON	REPLACES	ORPERSON	PATTERN	-9-9-9																								
Required name																												

Figure 5-17 Defining a Transaction

In the top portion of the window, there are two transaction identifier lines. The entries in these lines are used to identify the transaction record. The example uses the identifiers ID1 and ID2; they are combined as one identifier for a transaction that adds orders to the CUSTMAST master file.

Each identifier consists of three entries: Name (field name in the transaction file), Oper (operator), and Code (the value used to identify the transaction). In the example, both operator entries test for an equal condition. The contents of the transaction field ID1 must be equal to ADD and the contents of ID2 must be equal to ORDR for the transaction record to be applied.

The lower portion of the window specifies the actions to be taken against the master file. In the example, the action code against CUSTNO is UPDATE/INSERT. This means that the record on the transaction file is inserted as a new record on the master file if a matching CUSTNO record is not found. If a match on CUSTNO occurs, the update action is performed. Once a match occurs on CUSTNO or a new record is created, the remaining fields and 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 action code indicates that the segment is to be inserted into the master file. If a match occurs, the fields following it are processed.

The REPLACES action against ORDRDATE replaces the master file field with the transaction field.

The ORPERSON field is validated by a pattern of -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.

Creating VISION:Builder Applications

Application Development Windows

This chapter describes the use of the VISION:Builder Application Development component of VISION:Workbench. It is used to define the files, transaction groups, procedures, requests, reports, and parameters used by your VISION:Builder application. You enter it by selecting Builder under the Application Development heading on the Main Menu. [Figure 6-1](#) shows the structure of how the windows within the VISION:Builder application development component are organized.

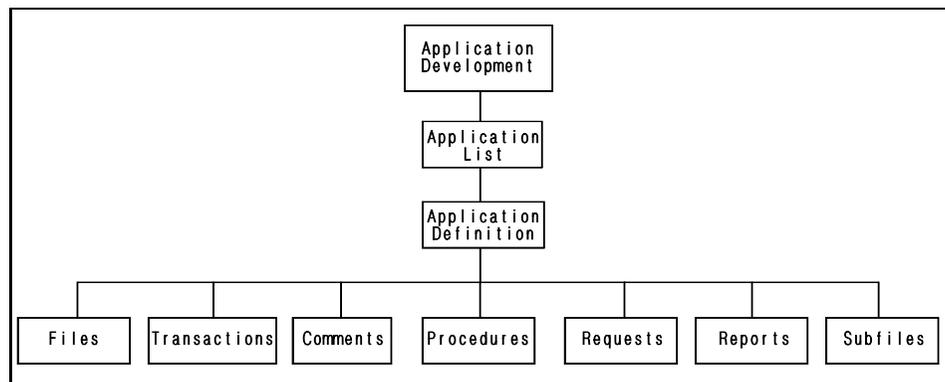


Figure 6-1 VISION:Builder Application Development Component

VISION:Builder Application List Window

The VISION:Builder Application List window, shown in [Figure 6-2](#), is the first window opened after the Main Menu. This window lists the name of all existing VISION:Builder applications.

You select an existing application to be modified or submitted by placing your cursor on it and pressing F8. You can add new application names to the list at any time. When you add a name to the list or select an existing one, the Application Definition window, shown in [Figure 6-3](#), opens.

VISION:Workbench for DOS		Current :
USERINFO	User file index information	FI0100
Type	Name	Location
BUILDER	EX1	C:\MPCXX\SAMPAPPS
BUILDER	EXAMPLE	C:\MPCXX\SAMPAPPS
BUILDER	OLYMPICS	C:\MPCXX\SAMPAPPS
BUILDER	TRANRUN	C:\MPCXX\SAMPAPPS
Name of item: press F8 to select this item.		

Figure 6-2 VISION:Builder Application List Window

Application Definition Window

The Application Definition window is used to specify the various parts of the application. From the selections made on the Application Definition window, some or all of the following windows open and are used to complete the application definition:

- Files Used
- Application Parameters
- Subfile Definitions
- Requests
- Transaction Groups Used
- Comments for Applications
- Procedure Definitions
- Report Definitions

As shown in [Figure 6-3](#), the Application Definition window contains a window where you can enter information on the control procedures for each of the primary control events within the VISION:Builder cycle. These events correspond to the traditional VISION:Builder request types (for example, N, E, 1, 2, M, 3, and 4).



Figure 6-3 Application Definition Window

With this concept, you can specify only one procedure for each of these events. All other procedures and all requests are automatically designated as subroutine procedures or requests, with the exception of preselection procedures or requests. You specify preselection procedures and requests by entering a Y in the Pre-selection entry on the procedure and request windows.

If the name of an event controlled procedure on the window is changed, the old procedure type is changed to a subroutine and the new procedure becomes the appropriate type. If the name of an existing event controlled procedure on the window is blanked out, the procedure becomes a subroutine and there is no event controlled procedure for the corresponding type.

When you have finished entering information on the Event Controlled Procedures window, press F3. You can reopen the window by positioning the cursor on the button field Control and pressing F8.

The Application Definition window is divided into two functional parts. The top part shows a numeric value next to each entry with the exception of Control and Params (they have no count), and Report Mgr and Comments (they display YES or NO).

The entries on the first line are button fields that are used to open windows and to maintain a count of that application object. You open the window by placing the cursor over the button field and pressing F8. A window appropriate to the button field opens. The number of specifications made in the window is reflected in the count fields on the Application Definition window (for example, if three files are defined in the files window, the Files entry on the Application Definition window contains a 3).

You cannot make selections on the second line of the window, except for Report Mgr. These entries are not button fields and are controlled by defining procedures, reports, requests, and subfiles on the lower portion of the window. The number of items defined is reflected in Procs, Requests, Reports, and Subfiles on the second line.

The bottom portion of the window is used as a list to describe the type of application object and to assign it a name. The valid application definition types are Procs, Reports, Requests, and Subfiles. When you select an existing entry from the list or make a new one and select the appropriate menu bar item, the appropriate definition window opens. [Figure 6-4](#) shows a completed Application Definition window.

VISION:Workbench for DOS		Working:EX1	
X=EXPORT		RETRIEVE	
Application Name: <input type="text" value="EXAMPLE"/>			RC4000
Builder Application Sections			
Files.....:	1	Control	Params
Report Mgr:No		Procs....:	1
		Requests:	0
		Reports.:	1
		Subfiles:	0
		Transact:	0
		Comments:	No
Type	Name	Description	
PROC	VALUE	;EACH MASTER-FILE RECORD CONTROL PROCEDU	
REPORT	VALUEREP	REPORT SHOWING VALUE OF INVENTORY	
Required Name			

Figure 6-4 Completed Application Definition Window

Files Used Window

The Files Used window, shown in [Figure 6-5](#), opens when you select the Files button field on the Application Definition window. The Files Used window contains the names of the files used in the application and describes their processing characteristics. The number of files specified on this window is reflected in the Files button field count on the Application Definition window.

VISION:Workbench for DOS				Working:EXAMPLE			
COPY MOVE SEARCH REPLACE				Application Name:EXAMPLE RC4100			
File description			Alternate	A I C D	User I/O	Password/	U D C
Qual Name	DD-Name	Lbl	Key(s)	r C r B	Module	Auth.-ID	p i h
				r F d I			d r k
ITEM	M4OLD	S					

Qualifier:0,N,X,1-9,A,B,E,G,H,J,K,M,Q,W,U

Figure 6-5 Files Used Window

Entries on the Files Used window fall into two functional types: file description entries, and run time entries. The entries in the leftmost third of the window are the file description entries. The run time entries display in the other two-thirds of the window. You move from entry to entry with your cursor or by filling the entire contents of the entry; in which case, the cursor is automatically positioned on the next entry. Online help is available for each of the entries in this window.

Application Parameters Window

The Application Parameters window opens when you select the Params button field on the Application Definition window. The Application Parameters window is used to specify processing, report, and run time parameters. [Figure 6-6](#) shows a sample of this full-screen window. You can enter these entries for the first time or change them by positioning the cursor on the entry and keying in the appropriate information.

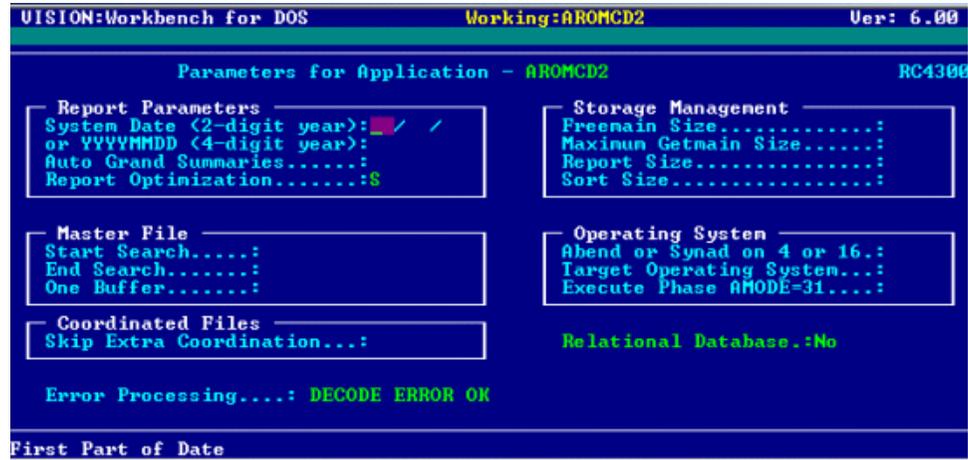


Figure 6-6 Application Parameters Window

Transaction Groups Used Window

The Transaction Groups Used window opens when you select the Transact button field on the Application Definition window. The Transaction Groups Used window is used to specify the transaction groups used in the application. [Figure 6-7](#) shows a sample Transaction Groups Used window.

VISION:Workbench for DOS		Working:EXAMPLE								
COPY MOVE SEARCH REPLACE										
Application Name:EXAMPLE			RC4000							
Builder Application Sections										
Files.....:	1	Control	Params							
Report Mgr:No	Procs...:	1	Requests: 0							
			Reports.: 1							
			Subfiles: 0							
Type	Name	Description								
PROC	VALUE	;EACH MASTER-FILE RECORD CONTROL PROCEDU								
REPORT	VALUEREP	REPORT SHOWING VALUE OF INVENTORY								
<table border="1"> <tr> <td colspan="2" style="text-align: center;">RC4200</td> </tr> <tr> <td colspan="2" style="text-align: center;">Transaction Groups</td> </tr> <tr> <td colspan="2" style="text-align: center;"> <table border="1"> <tr> <td style="text-align: center;">ORDENTRY</td> </tr> </table> </td> </tr> </table>				RC4200		Transaction Groups		<table border="1"> <tr> <td style="text-align: center;">ORDENTRY</td> </tr> </table>		ORDENTRY
RC4200										
Transaction Groups										
<table border="1"> <tr> <td style="text-align: center;">ORDENTRY</td> </tr> </table>		ORDENTRY								
ORDENTRY										
Transaction Group Name										

Figure 6-7 Transaction Groups Used Window

Each transaction group name you specify must be previously defined, validated, and saved in the VISION:Workbench library. (See [Defining VISION:Builder Transactions on page 5-18](#) for more information.) Each transaction group name entered on this window is reflected in the count of the Transact entry on the Application Definition window.

Application Comments Window

The Comments window opens when you select the Comments button field on the Application Definition window. The Comments window is used to specify comments or descriptive information about the application. You can enter them in the window in any format you like.

Subfile Definition Window

The Subfile Definition window is used to specify the file characteristics of the subfile and list the fields to be contained in the subfile. To open the Subfile Definition window, either select an existing subfile or enter a new one. To enter a new subfile, type an S in the Type entry (the rest of the word is filled in for you), or press F6 for Choices and make your selection. Enter the name of the subfile, a description, and press F8. [Figure 6-8](#) shows a sample of this window.

VISION:Workbench for DOS		Working:EXAMPLE	
COPY MOVE SEARCH REPLACE			
Destination Name:SUBFILE1		Subfile Name:ITEMSUB	SF0100
Data Selection.:ALL		Comments.....: No	
		Output Characteristics: No	
Field Name	Field Name	Field Name	
ITEMNO			
ITEMNAME			
ITMPRICE			
Required Name			

Figure 6-8 Subfile Definition Window

To create a subfile, you must also have the files specified on the Files Used window, as well as have a procedure defined that calls the subfile. [Figure 6-9](#) and [Figure 6-10](#) show samples of these windows.

VISION:Workbench for DOS		Working:EXAMPLE	
COPY MOVE SEARCH REPLACE			
Application Name:EXAMPLE		RC4100	
Qual	File description	Alternate	User I/O Password/
	Name DD-Name Lbl	Key(s) r C r B	Module Auth.-ID
		r F d I	U D C
			p i h
			d r k
	ITEM M4OLD S		
	SUBFILE1 M4SUBF1 S		
Qualifier:0,N,X,1-9,A,B,E,G,H,J,K,M,Q,W,U			

Figure 6-9 Files Used Window with Subfile Specified

VISION:Workbench for DOS		Working:EXAMPLE
COPY	MOVE	SEARCH REPLACE
Procedure Name:SUBPROC	Type : S	RQ0100
Pre-Selection : N	Re-init Temps:	Maximum Items Selected:
<pre> IF QANONHND EQ 5 CALL SUBFILE ITEMSUB END </pre>		
Enter a Procedure Statement		

Figure 6-10 Procedure Calling a Subfile

Procedure Definition Window

To open the Procedure Definition window, either select an existing procedure or enter a new one. To enter a new procedure, type a P in the Type entry (the rest of the word is filled in for you), or press F6 for Choices and make your selection, or leave the Type entry blank (procedure is the default). Enter the name of the procedure, a description, and press F8. [Figure 6-11](#) shows a sample of this window.

The Procedure Definition window consists of a header section that specifies procedure processing requirements and a detail section below the header section. You use the detail section to enter ASL statements. The ASL statements are described in the ASL Reference Guide.

You must define exactly where reports are to be used in your application. You do this by calling the report in the application processing procedure. See [Chapter 7, Creating Reports](#) for more information on defining reports.

VISION:Workbench for DOS		Working:EXAMPLE	
COPY MOVE SEARCH REPLACE			
Procedure Name	<input type="text" value="VALUE"/>	Type : N	RQ0100
Pre-Selection :	N	Re-init Temps:	Maximum Items Selected:
;EACH MASTER-FILE RECORD CONTROL PROCEDURE			
;			
STOCKVAL: FIELD P 5 DEC 2			
IF QANONHND > 0			
LET T.STOCKVAL = QANONHND * UNITCOST			
CALL REPORT VALUERE			
END			
Required Name			

Figure 6-11 Procedure Definition Window

Request Definition Window

To open the Request Definition window, either select an existing request or enter a new one. To enter a new request, type REQ in the Type entry (the rest of the word is filled in for you) or press F6 for Choices and make your selection. Enter the name of the request, a description, and press F8.

You can use the Request Definition window in place of the Procedure Definition window which uses the free-form procedural language, ASL. On the Request Definition window, you can enter requests consisting of the fixed format PR and TF statements. A Request Definition window is shown in [Figure 6-12](#).

VISION:Workbench for DOS										Working:EXAMPLE		
COPY MOVE SEARCH REPLACE												
Request Section - REQ1										Type :	S	PR4000
Temp. Fields.: Yes					Pre-Selection...: N					Max Item Select.:		
Re-init Temps:					Back Branch Ctl.:					Comments.....: No		
Seq	L	C	Q	Operand	O	Q	Operand	Q	Operand	Partial		
/	U	O	L	...A....	P	L	...B....	L	...C....	S	N	O
Lbl	L	N	F	Name	R	F	Name/Constant	F	Name	T	O	P
				QANONHND	GT	D	0					
				QANONHND	*	UNITCOST		T	STOCKVAL			
Request Name												

Figure 6-12 Request Definition Window

Temporary fields defined by either procedures or requests are global; any procedure, request, report, or subfile can reference them. The Request Definition window does not support the GSI CALL format. You can only code GSI calls from the Procedure Definition window.

To define temporary fields from this window, put the cursor on the Temp. Fields button field and press F8. This opens the Temporary Field Definition window where you can define temporary fields. You can enter up to two lines of column heading text for each field through a pop-up window on this window. Press F3 to close the window.

Report Definition Window

The Report Definition window is used to describe the format and contents of the report. To open the Report Definition window, either select an existing report or enter a new one. To enter a new report, type REP in the Type entry (the rest of the word is filled in for you) or press F6 for Choices and make your selection. Enter the name of the report, a description, and press F8. [Figure 6-13](#) shows a Report Definition window.

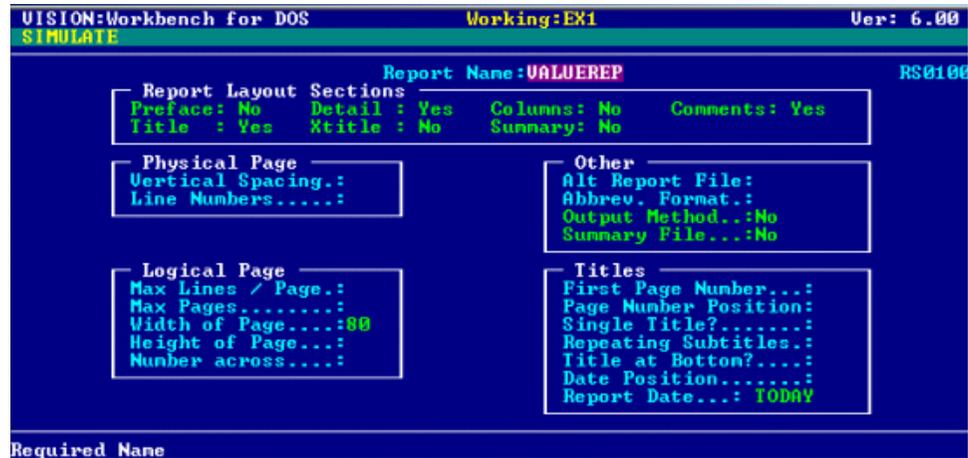


Figure 6-13 Report Definition Window

You define where your reports are to be used in the application using CALL statements on the Procedure Definition window. See [Chapter 7, Creating Reports](#) for more detailed information about defining reports.

Report Manager Window

The Report Manager window is used to specify the report collating and routing specifications for the application. To open the Report Manager window, select the Report Mgr button field on the Application Definition window. You must assign the window a name following the standard naming conventions. The body of the window may contain COLLATE and ROUTE statements, along with the appropriate comments using ASL conventions. (See the ASL Reference Guide for a description of these statements.) [Figure 6-14](#) shows an example of this window.

VISION:Workbench for DOS		Working:RMEXAMPL	
COPY MOVE SEARCH REPLACE			
Application Name:RMEXAMPL			RC4000
Builder Application Sections			
Files.....:	1	Control	Params
Report Mgr:Yes	Procs....:	1	Requests: 0
			Transact: 0
			Comments:No
			Reports.: 3
			Subfiles: 0
Type	Name	Description	
PROC	MAIN	;EACH MASTER-FILE-RECORD CONTROL PROCEDU	
REPORT	REP1		
REPORT	REP2		
REPORT	REP3		
Report Manager Group Name:REPMGR			
; Print reports in reverse order			
COLLATE REPORTS REP3 REP2 REP1			
; Route Report REP1 to DESTX, Reports REP2 and REP3 to DESTY			
ROUTE REPORT REP1 TO DESTX			
ROUTE REPORTS REP2 REP3 TO DESTY			
Enter name of Report Manager statement group			

Figure 6-14 Report Manager Window

Retrieving Application Objects

You can use existing reports, subfiles, procedures, and requests to create new applications or to modify existing ones. The RETRIEVE menu bar item on the Application Definition window is available for this purpose. This menu item allows you to select reports, subfiles, procedures, or requests that are already defined and stored in the VISION:Workbench library. Use the RETRIEVE menu item as follows:

1. Press F4 to move the cursor to the EXPORT and RETRIEVE menu items (see [Figure 6-15](#)).

VISION:Workbench for DOS		Working:EXAMPLE	
X=EXPORT		RETRIEVE	
Application Name:EXAMPLE			RC4000
Builder Application Sections			
Files.....:	2	Control	Params
Report Mgr:No		Procs...:	1
		Requests:	0
		Transact:	1
		Comments:	No
		Reports.:	1
		Subfiles:	0
Type	Name	Description	
PROC	VALUE	;EACH MASTER-FILE RECORD CONTROL PROCEDU	
REPORT	VALUEREP	REPORT SHOWING VALUE OF INVENTORY	
Retrieve an item			

Figure 6-15 EXPORT and RETRIEVE Menu Bar

2. To select RETRIEVE, place the cursor on it and press Enter. The menu bar items display: SUBFILE, REPORT, PROCEDURE, and REQUEST (see [Figure 6-16](#)).

VISION:Workbench for DOS		Working:EXAMPLE	
SUBFILE	REPORT	PROCEDURE	Q=REQUEST
Application Name:EXAMPLE			RC4000
- Builder Application Sections -			
Files.....:	2	Control	Params
Report Mgr:No	Procs....:	1	Requests: 0
			Transact: 1
			Comments: No
			Reports.: 1
			Subfiles: 0
Type	Name	Description	
PROC	VALUE	;EACH MASTER-FILE RECORD CONTROL PROCEDU	
REPORT	VALUEREP	REPORT SHOWING VALUE OF INVENTORY	
Operate on procedure specification			

Figure 6-16 SUBFILE, REPORT, PROCEDURE, and REQUEST Menu Bar

3. Select SUBFILE, REPORT, PROCEDURE, or REQUEST by placing the cursor on one and pressing Enter. A list of names for the selected application object displays at the bottom of the window (see [Figure 6-17](#)).

VISION:Workbench for DOS		Working:EXAMPLE	
X=EXPORT	RETRIEVE		
Application Name:EXAMPLE			RC4000
- Builder Application Sections -			
Files.....:	1	Control	Params
Report Mgr:No	Procs....:	1	Requests: 0
			Transact: 0
			Comments: No
			Reports.: 1
			Subfiles: 0
Type	Name	Description	
PROC	VALUE	;EACH MASTER-FILE RECORD CONTROL PROCEDU	
REPORT	VALUEREP	REPORT SHOWING VALUE OF INVENTORY	
↑ * ← Esc F3			
BUILD	PRINT	PRNTCNTL	TOTAL
Select procedure to retrieve with arrow keys; then ↓			

Figure 6-17 Application Object List

- Select the application object you want and press Enter. The selected application object's type and name are inserted on the Application Definition window (see [Figure 6-18](#)). All related specification windows are included in the application.

VISION:Workbench for DOS		Working:EXAMPLE	
X=EXPORT RETRIEVE			
Application Name:EXAMPLE			RC4000
Builder Application Sections			
Files.....: 1	Control	Params	Transact: 0
Report Mgr:No	Procs...: 2	Requests: 0	Comments: No
		Reports.: 1	Subfiles: 0
Type	Name	Description	
PROC	VALUE	:EACH MASTER-FILE RECORD CONTROL PROCEDU	
PROC	TOTAL	:SUM THE ITEMS IN INVENTORY	
REPORT	VALUEREP	REPORT SHOWING VALUE OF INVENTORY	
Required Name			

Figure 6-18 Retrieved Application Objects

- You can select, modify, or use retrieved application objects as is.
- You must save (F5) the entire application, including the retrieved objects, to make a permanent copy of the application.

Report Definition Windows

This chapter describes defining VISION:Builder reports using VISION:Workbench. Report definitions are a function of the VISION:Builder Application Development component. You use the Report Definition windows to paint reports, to specify report characteristics, and to simulate how the report output is to look. With VISION:Workbench, you can define both standard reports and painted reports.

You open the Report Definition windows by selecting a report entry from the list of names on the lower portion of the Application Definition window. [Figure 7-1](#) shows the structure of how the Report Definition windows are organized.

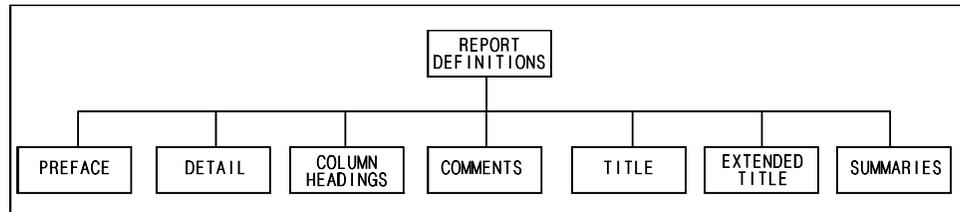


Figure 7-1 Report Definition Windows

Report Definition Window

The Report Definition window, shown in [Figure 7-2](#), is the first window to open after you select a report on the Application Definition window. You can use the Report Definition window to create a new report or to modify an existing one.

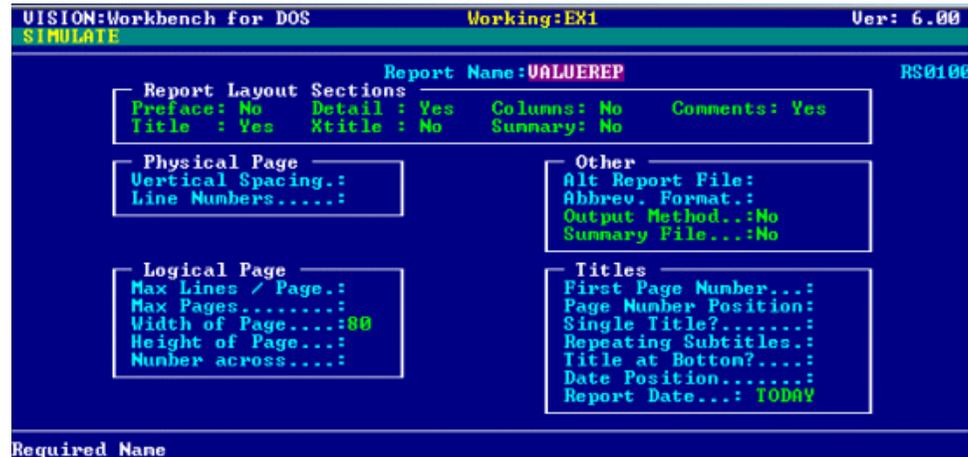


Figure 7-2 Report Definition Window

The top portion of the window (the Report Layout Section) contains button fields that you can select. When you select a button field (by placing the cursor on it and pressing F8), a window for painting and specifying that section of the report opens. Once you have painted the report section and return to the Report Definition window, the value of the entry you selected on the Report Definition window changes from NO to YES (for example, Comment=YES).

The lower portion of the Report Definition window contains entries for formatting reports.

Some or all of the following windows are used to complete report definitions for standard reports and painted reports:

- Report Preface
- Report Details
- Report Column Headings
- Report Comments
- Report Titles
- Report Extended Titles
- Report Summaries

You create painted reports by selecting any or all of the following:

- Report Column Headings
- Report Extended Titles
- Report Summaries

Report Preface Window

The Report Preface window allows you to paint a preface section for a report. Key in the preface information as you want it to appear in your preface. You can use the arrow keys to move the cursor to whatever location you want. The Line and Col indicators at the top of the window indicate the line and column number that the cursor is on. A Report Preface window is shown in [Figure 7-3](#).

VISION:Workbench for DOS				Working:EXAMPLE	
CENTER				EDIT	
Line	1	Col	1	Report Preface - VALUERE	RS0120
<p>THIS IS THE PREFACE PAGE FOR THE REPORT 'VALUERE'</p> <p>THIS REPORT SOULD BE RUN MONTHLY AND GIVEN TO THE VICE PRESIDENT OF SALES</p>					

Figure 7-3 Report Preface Window

When you paint preface lines wider than the window, the display scrolls horizontally. When the page width is narrower than the window, a double vertical line is used to mark the right margin.

The CENTER and EDIT menu bar is available on this window and works the same as described in [Chapter 5, Entering Definitions](#).

Report Details Window

The Report Details window is used to specify the fields to be contained in the report. It also specifies how the report is to be formatted and where control breaks and totals are to be taken. A Report Details window is shown in [Figure 7-4](#).

Working:EXAMPLE										
COPY	MOVE	SEARCH	REPLACE							
Detail Section - VALUEREP									RS0110	
Summaries.....										
T L D										
Field Name	Sp	End	Non	Sort...	Break..	y	v	e	%Ratio	Output
	Bf	Lin	Prt	Seq/Dec	Ctl/Sub	p	l	c	Denominatr	Edit
ITEMNO										
UNITCOST									X G	
									M G	
QANONHND										
ITMPRICE										
T.STOCKVAL				1					T G	
Field Name										

Figure 7-4 Report Details Window

To specify multiple summaries on a field, enter the additional summary on the line beneath the field, without repeating the field name. This is shown in [Figure 7-4](#) on the field UNITCOST, where both maximum (X) and minimum (M) grand summaries are taken.

Report Column Headings Window

The Report Column Headings window is used for specifying standard reporting, as well as specifying painted reporting. A Report Column Headings window is shown in [Figure 7-5](#).

VISION:Workbench for DOS				Working:EXAMPLE	
CENTER	GEN	EDIT	Y=CONTENTS		
Line 1	Col 72	Column Heading Section - VALUEREP			RS0210
Col Heading Type.....: Col Heading Position.: Col Heading Character:*					
DATE _____				PAGE _____	

ITEM	ITEM	QUANTITY	ITEM	STOCKVAL	
NUMBER	COST	ON HAND	PRICE		

<div style="border: 1px solid black; padding: 5px; width: fit-content;"> Field Name F.TODAY F.PAGE </div>					
Field Name					

Figure 7-5 Report Column Headings Window

The top portion of the window (up to the Column Heading Character) is used for both types of reporting. The lower portion of the window is used only for painted reports.

You use the top portion of the window to make changes to column headings defined on the Segment Definition window. (See [Chapter 4, Using the System](#) for more detailed information about defining column headings in definitions.)

The entries in the top portion of the window allow you to:

- Use the field name instead of the column heading.
- Completely suppress the printing of column headings.
- Print column headings at the bottom of the column instead of at the top.
- Change column heading borders.

You use the lower portion of the window (below the column heading entries) for creating column headings in painted reports. You can paint column headings the way you want them to appear in your report. You can place variable information in the column headings by placing underscores where you want variable information to appear. The contents of variable fields are filled by the field names entered in the contents window of the window.

You open the Contents window by pressing F4 and selecting the CONTENTS menu bar item. The Contents window opens on the bottom part of the screen. In this window, you list fields that provide values to the variable information (underscored areas). Under the Field Name entry, enter the names of the fields that fill the variables.

Each field name specified on a line is directly related to its underscored variable in the top half of the window. The first line is associated with the first variable, the second line, to the second variable, and so on, counting left to right, top to bottom.

Any associated editing information is also defined in the lower window of the window.

You can select the GEN menu bar item to help you see where the column headings are to be placed in the report. (See [Figure 7-5](#) for a sample.) This menu item displays any column headings defined in the file definition. Once displayed, you can rearrange, delete, or change them.

Report Comments Window

The Report Comments window is used to include descriptive information about the report. You key in any descriptive information that you want to be saved with your preface. You can use the arrow keys to move the cursor to whatever location you want. These comments are stored with the application and transported with it when the application is sent to the host.

The COPY, MOVE, SEARCH, and REPLACE menu bar is available on this window and works the same as described in [Chapter 4, Using the System](#). A Report Comments Window is shown in [Figure 7-6](#).

VISION:Workbench for DOS	Working:EXAMPLE
COPY MOVE SEARCH REPLACE	
Comments for : Report	VALUEREP CM0100
REPORT SHOWING VALUE OF INVENTORY	
Enter a comment line	

Figure 7-6 Report Comments Window

Report Title Window

The Report Title window is used to paint titles in reports. The page title is painted on the window as it is to appear in the report. A Report Title window is shown in [Figure 7-7](#).

VISION:Workbench for DOS			Working:EXAMPLE		
CENTER EDIT					
Line	1	Col	1	Report Title - VALUEREP	RS0130
REPORT TITLE - VALUEREP REPORT FOR ABC COMPANY REPORT OF THOSE ITEMS THAT ARE IN STOCK AND THE VALUE OF THE INVENTORY					

Figure 7-7 Report Title Window

You can use the arrow keys to move the cursor to whatever location you want. The Line and Col indicator at the top of the window indicates the line and column number that the cursor is on.

When you paint a title line wider than the window, the display scrolls horizontally. When the page width is narrower than the window, a double vertical line is used to mark the right margin.

The CENTER and EDIT menu bar is available on this window and works the same as described in [Chapter 5, Entering Definitions](#).

Report Extended Titles Window

The Report Extended Titles window is used to create titles in painted reports. Titles painted on this window replace titles painted on the Report Title window. A Report Extended Titles window is shown in [Figure 7-8](#).

VISION:Workbench for DOS				Working:EXAMPLE				
CENTER GEN EDIT Y=CONTENTS								
Line	1	Col	74	Page Title Section - VALUEREP			RS0200	
				REPORT TITLE VALUEREP		PAGE _____		
				REPORT FOR ABC COMPANY				
				REPORT OF THOSE ITEMS THAT ARE IN STOCK AND				
				THE VALUE OF THE INVENTORY				
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 5px;"> Field Name F.TODAY F.PAGE </td> </tr> </table>								Field Name F.TODAY F.PAGE
Field Name F.TODAY F.PAGE								
Field Name								

Figure 7-8 Report Extended Titles Window

On the Report Extended Titles window, you can paint titles the way you want them to appear in your report. You can place variable information anywhere in the title by placing underscores where you want the variable information to appear.

The contents of variable fields are filled by the field names entered in the contents window. You open the contents window by pressing F4 and selecting the CONTENTS menu bar item.

The contents window opens on the bottom part of the screen. In this window, you list fields that provide values to the variable information (underscored areas).

Each field name specified on a line is directly related to its underscored variable in the top half of the window. The first line is associated with the first variable, the second line, with the second variable, and so on, counting left to right, top to bottom.

You can select the GEN menu bar item to help you see where the headings are to be placed in the report, as shown in [Figure 7-8](#).

Using the Report Definition Menu Bar

The report definition menu bar displays on the second line of each window. To select the menu bar item you want, you press F4 and either move the cursor to it and press Enter or enter its first letter (for example, enter a C to select the CENTER menu item). The following menu bar items are available on the report definition windows:

- CENTER
- GEN
- EDIT
- CONTENTS
- LEVEL
- SIMULATE

Not all menu bar items listed are available on every Report Definition window. Each window displays which menu bar items are available.

The CENTER and EDIT menu bar items work the same in report definitions as they do in file definitions. (See [Chapter 5, *Entering Definitions*](#) for a complete description of CENTER and EDIT.) When you select EDIT, the COPY, MOVE, SEARCH, and REPLACE menu bar displays. (See [Chapter 4, *Using the System*](#) for a complete description of COPY, MOVE, SEARCH, and REPLACE.)

GEN Menu Bar Item

The GEN menu bar item is used for painted reports. Select it to paint column headings, titles, and summary lines. On the Report Summaries window, select GEN to display a default summary section that you can edit to produce a customized summary section. It displays column headings and titles on the Column Headings and Report Extended Titles windows.

CONTENTS Menu Bar Item

The CONTENTS menu bar item is used to specify fields that provide values to variable locations (underscores) in painted reports. When you select this menu item, a window opens on the bottom portion of the screen. Under the Field Name entry displayed in this window, you can enter any number of field names, arithmetic expressions (report summary definitions only), or partial field functions. These entries fill the variables specified on the top part of the window. An example of the CONTENTS window is shown in [Figure 7-9](#).

LEVEL Menu Bar Item

The LEVEL menu bar item is used on the Report Summaries window to specify the control break levels to print painted lines. The control break number precedes and follows the painted line.

SIMULATE Menu Bar Item

The SIMULATE menu bar item is used to produce a sample of a defined report. The sample includes sample values for all fields and summaries.

Simulating Reports

The Report Definition component lets you see what the reports that you define look like. After you define it, you can display a sample report. Press F4 on the Report Definition window to move the cursor to the SIMULATE menu item and press Enter. A sample report, including sample values for all fields and summaries, displays on the window. [Figure 7-10](#) shows a sample report generated by SIMULATE.

VISION:Workbench for DOS		Working:EXAMPLE	
Line	2	Col	1
Report Simulation - VALUEREP			RS0300
01/15/95	REPORT TITLE - UNIT COST ITEM VALUE REPORT		PAGE 1
REPORT FOR ABC COMPANY			
REPORT OF THOSE ITEMS THAT ARE IN STOCK AND			
THE VALUE OF THE INVENTORY			
DATE	_____	PAGE _____	

ITEM	ITEM	QUANTITY	ITEM
NUMBER	COST	ON HAND	PRICE

AAAAAAA	\$.01	1	\$.01
ZZZZZZZ	9,999.99	99,999	9,999.99
GRAND TOTAL			999.99
MAX.			_____
MIN.			_____

Figure 7-10 Simulated Report

Walk-Through for VISION:Builder Applications

This chapter lets you get a feel for VISION:Workbench by actually using it. A simple VISION:Builder application has been predefined in the library. The changes you are asked to make to the application may not seem realistic; they are only intended for you to use as much of VISION:Workbench as possible.

For the best results, enter the steps of this walk-through on your PC. However, this chapter contains copies of every window and PC interaction and can be read without the use of a PC.

As you work your way through the application, you select menu items, enter information, use the function keys, and use the menu bar. For each step of the way, the action to be performed in the particular window is described and an illustration of the window is included both before and after the action is taken. If the action results in another window being opened, that window is shown instead. Your window should always resemble the illustrations in this chapter.

Throughout most of this walk-through, the window you are looking at on your PC is shown on the top half of the page; the action you must take is on the bottom half of the page.

The walk-through concludes by having you export the application. Once exported, the application can be uploaded to the host where it can be subsequently processed.

Application Overview

The application in this chapter processes an input file that is used to produce a report. The file is an inventory file that contains activity summaries such as quantity on order, year to date sales, amount backordered, and quantity on hand. It also includes control information such as reorder points, minimum orders, expected costs, and prices.

The application reads the inventory file and produces a report showing the value of all inventory in stock. You are asked to modify the application to incorporate a new input file containing new data and fields. Because of the new data, there is an additional reporting requirement. You must change the file definition, application definition, report definition, and procedural logic in the application.

Refer to the examples and their accompanying text as you go through the exercise. This example assumes that you are a knowledgeable VISION:Builder user; therefore, VISION:Builder entries are not described.

Important Keys

There are several keys that you should be familiar with before beginning the sample application. The following keys and function key descriptions are used for navigation and operation of the sample application. See [Chapter 4, Using the System](#) for a complete discussion on using the system.

Function Key	Description
Enter (Return)	Allows you to navigate through windows or select an item from a Choices list.
F1	Displays the function key bar which provides you with the meaning of the available function keys for the current window.
F3	Indicates that you are finished with a window. It closes the window and takes you back to the higher level window. It also terminates processing when pressed from the Main Menu.
F4	Moves the cursor to the menu bar which displays the menu items available for the current window.
F8	Opens a window where you can provide or display more information on the current item.

Beginning the Application

This example asks you to do the following:

- Update an existing application to use a new file.
- Change a field name in a file definition.
- Update a report definition to include a new field.
- Add the new field name to the procedural definition.
- Update the application definition to incorporate the new file.

Turn your PC on and go to the directory containing the VISION:Workbench for DOS.

Type mpcx and press Enter.

The Primary Library Path on the Setup Window must be specified as \MPC55\SAMPAPPS to access the sample application. See Specifying Setup Information in [Chapter 2, Getting Started – Installation and Setup](#) for information on the Setup Window.

The Banner window appears first. Press any key to move to the VISION:Workbench for DOS Copyright window (not shown). Press any key again to move to the Main Menu.

VISION:Workbench for DOS										Current :
Select an activity :										MAIN
<div style="border: 1px solid black; padding: 5px; display: inline-block;"> Application Development Builder O = Transact </div>										
<div style="border: 1px solid black; padding: 5px; display: inline-block;"> Definitions Files Tables Arrays X = Transactions </div>										
<div style="border: 1px solid black; padding: 5px; display: inline-block;"> Miscellaneous Utilities Setup Parameters </div>										
F1 Help	F2 Names	F3 Quit	F4	F5	F6	F7	F8	F9	Alt+F10	
Press ENTER or F8 to select this item										

This is the Main Menu. Start by changing a file definition.

To select the file definition activity, move your cursor to Files and press F8 or simply press F for Files.

Follow the instructions on the bottom line of the screen (watch the bottom line; it always provides a helpful prompt).

VISION:Workbench for DOS		Working :	
USERINFO	User file index information	FI0100	
Type	Name	Location	
FILE	CUSTOMAST	C:\MPCXX\SAMPAPPS	
FILE	CUSTOMER	C:\MPCXX\SAMPAPPS	
FILE	DEPTFILE	C:\MPCXX\SAMPAPPS	
FILE	ITEM	C:\MPCXX\SAMPAPPS	
FILE	ITEMASTX	C:\MPCXX\SAMPAPPS	
FILE	ITEMX	C:\MPCXX\SAMPAPPS	
FILE	OLYMRSLT	C:\MPCXX\SAMPAPPS	
FILE	PERSFILE	C:\MPCXX\SAMPAPPS	
FILE	TRANWORK	C:\MPCXX\SAMPAPPS	
FILE	WORKSTOR	C:\MPCXX\SAMPAPPS	
Name of item; press F8 to select this item.			

You now have a list of file definitions on the File List window.

Place the cursor on the ITEM file definition and then press F8.

VISION:Workbench for DOS		Working: ITEM	
L=RELATION	ABOVE	ORPHANS	RETRIEVE DELETE X=EXPORT
File Name:	ITEM	File Identifier:	Comments: No FD0100
Record Format:	VARIABLE	Record Size:	Records Per Block: Buffer Size: 3120
ITEMSEG			
Required Name			

This is the file structure (single-level file) on the File Definition window.

Create a copy of the application by simply changing the name from ITEM to ITEMAS.

All future actions will modify ITEMAS and leave the original ITEM undisturbed.

VISION:Workbench for DOS		Working:ITEM			
L=RELATION	ABOVE	ORPHANS	RETRIEVE	DELETE	X=EXPORT
File Name:ITEMAS	File Identifier:	Comments: No	FD0100		
Record Format:VARIABLE	Record Size:	Records Per Block:	Buffer Size: 3120		
<div style="border: 1px solid black; padding: 2px; display: inline-block;">ITEMSEG</div>					
Required Name					

This window shows the newly named file — ITEMAS. To look at the fields in this file, move your cursor to the ITEMSEG box and press F8.

VISION:Workbench for DOS		Working:ITEM		Ver: 6.00					
CENTER EDIT									
Segment Name.....:ITEMSEG					FD0110				
Occurs N Times.....:									
Key Sequence.....:									
Field Name	Type	Loc	Len	Dec	Rnd	Key	Count	More	Col
ITEMNO	CHARACTER	1	7			1			Y
ITEMNAME	CHARACTER	8	20						Y
ITEMPRICE	ZONED	28	6	2				Y	Y
ITEMCOST	ZONED	34	6	2				Y	Y
ITEMWGT	ZONED	40	3						Y
QANONHND	ZONED	43	5						Y
SHIPYTD	ZONED	48	6						Y
ORDQTY	ZONED	54	6						Y
QTYBKORD	ZONED	60	5						Y
REORDPT	ZONED	65	4						Y
REORDQTY	ZONED	69	4						Y
PRODCODE	CHARACTER	73	2						Y
VENOR	CHARACTER	75	4						Y
FILLER	FIXED POINT	79	1						
Required Name									

Now you see the fields in the segment on the Segment Definition window.

You want to change the ITMCOST field to UNITCOST, so move your cursor to ITMCOST and type UNITCOST over it.

VISION:Workbench for DOS		Working:ITEM		Ver: 6.00					
CENTER EDIT									
Segment Name.....:ITEMSEG					FD0110				
Occurs N Times.....:									
Key Sequence.....:									
Field Name	Type	Loc	Len	Dec	Rnd	Key	Count	More Attr	Col Hdg
ITEMNO	CHARACTER	1	7			1			Y
ITEMNAME	CHARACTER	8	20						Y
ITMPRICE	ZONED	28	6	2				Y	Y
UNITCOST	ZONED	34	6	2				Y	Y
ITEMWT	ZONED	40	3						Y
QANONIND	ZONED	43	5						Y
SHIPYTD	ZONED	48	6						Y
ORDQTY	ZONED	54	6						Y
QTYBKORD	ZONED	60	5						Y
REORDPT	ZONED	65	4						Y
REORDQTY	ZONED	69	4						Y
PRODCODE	CHARACTER	73	2						Y
VENDOR	CHARACTER	75	4						Y
FILLER	FIXED POINT	79	1						

Press F6 for choices

The name is changed; you are finished with this segment. (After entering the new name, UNITCOST, the cursor advanced to the next entry field, Type. Whenever the entire entry is filled, VISION:Workbench automatically skips to the next entry field.)

Press F3 to close this window and return to the previous window.

VISION:Workbench for DOS		Working:ITEMAST	
L=RELATION ABOVE ORPHANS RETRIEVE DELETE X=EXPORT			
File Name:	ITEMAST	File Identifier:	Comments: No FD0100
Record Format:	VARIABLE	Record Size:	Records Per Block: Buffer Size: 3120
ITEMSEG			
Required Name			

Now, you must validate the changed definition. To find out which function key will perform the validation, press F1 to display the function key bar.

VISION:Workbench for DOS				Working:ITEMAST					
L=RELATION ABOVE ORPHANS RETRIEVE DELETE X=EXPORT									
File Name:ITEMAST		File Identifier:		Comments: No		FD0100			
Record Format:VARIABLE				Record Size:		Records Per Block: Buffer Size: 3120			
ITEMSEG									
F1 Help	F2 Names	F3 Close	F4 Menu Bar	F5 Save	F6 Choices	F7 Validate	F8 Open	F9 Insert	Alt+F10 Delete

The function key bar displays across the bottom of the window.

Press F7 to validate the definition.

VISION:Workbench for DOS				Working:ITEMAST			
L=RELATION ABOVE ORPHANS RETRIEVE DELETE X=EXPORT							
File Name:ITEMAST		File Identifier:		Comments: No		FD0100	
Record Format:VARIABLE				Record Size:		Records Per Block: Buffer Size: 3120	
ITEMSEG							
Validation complete - no errors - Press any key.							

View the errors (in this case, there are none). Press any key to return to the previous window.

VISION:Workbench for DOS		Working: ITEMAST	
L=RELATION ABOVE ORPHANS RETRIEVE DELETE X=EXPORT			
File Name:	ITEMAST	File Identifier:	Comments: No FD0100
Record Format:	VARIABLE	Record Size:	Records Per Block: Buffer Size: 3120
ITEMSEG			
Required Name			

Now, you can SAVE the new definition in the VISION:Workbench library. Press F5, and the definition is saved. (Again, if you cannot remember which function key performs the SAVE, press F1 to display the function key bar.)

Press F3 to close this window and return to the previous window.

VISION:Workbench for DOS		Working: ITEMAST	
USERINFO		User file index information FI0100	
Type	Name	Location	
FILE	ITEMAST	Currently in memory	
FILE	CUSTOMAST	C:\MPCXX\SAMPAPPS	
FILE	CUSTOMER	C:\MPCXX\SAMPAPPS	
FILE	DEPTFILE	C:\MPCXX\SAMPAPPS	
FILE	ITEM	C:\MPCXX\SAMPAPPS	
FILE	ITEMASTX	C:\MPCXX\SAMPAPPS	
FILE	ITEMX	C:\MPCXX\SAMPAPPS	
FILE	OLYMRSLT	C:\MPCXX\SAMPAPPS	
FILE	PERSFILE	C:\MPCXX\SAMPAPPS	
FILE	TRANWORK	C:\MPCXX\SAMPAPPS	
FILE	WORKSTOR	C:\MPCXX\SAMPAPPS	
Name of item: press F8 to select this item.			

Now you see ITEM (provided for you) and ITEMAS (currently in memory - the copy that you just created).

Press F3 to return to the Main Menu.

VISION:Workbench for DOS Current:

Select an activity : MAIN

- Application Development
 - Builder** (highlighted)
 - U = Transact
- Definitions
 - Files
 - Tables
 - Arrays
 - X = Transactions
- Miscellaneous
 - Utilities
 - Setup
 - Parameters

F1 Help F2 Names F3 Quit F4 F5 F6 F7 F8 F9 Alt+F10

Press ENTER or F8 to select this item

To review the application definition, move the cursor to Builder and press Enter or F8. (You can also select an item from a menu by typing its first letter — in this case B.)

VISION:Workbench for DOS Current:

USERINFO User file index information FI0100

Type	Name	Location
BUILDER	EX1 (highlighted)	C:\MPCXX\SAMPAPPS
BUILDER	OLYMPICS	C:\MPCXX\SAMPAPPS
BUILDER	TRANRUN	C:\MPCXX\SAMPAPPS

Name of item; press F8 to select this item.

Now the Application List window displays.

Move your cursor to EX1 and press F8 to select it.



The Application Definition window appears with a pop-up window containing names of control procedures used in this application. The first five entries relate to initialization and record processing procedures. The last five entries relate to transaction processing procedures.

Note: This is equivalent to identifying a main line request which will call all other requests as subroutines. This concept is discussed in [Chapter 6, Creating VISION:Builder Applications](#).

Change the application name from EX1 to EXAMPLE by typing over it and pressing Enter. As with the file definition, you have just made an in-memory copy of the application and can now modify it. You can use this example again.

VISION:Workbench for DOS		Working:EX1	
X=EXPORT RETRIEVE			
Application Name:EXAMPLE			RC4000
Builder: Application Sections			
Files.....: 1	Control	Params	Transact: 0
Report Mgr:No	Procs...: 1	Requests: 0	Comments: No
		Reports.: 1	Subfiles: 0
Type	Name	Description	
PROC	VALUE	;EACH MASTER-FILE RECORD CONTROL PROCEDU	
REPORT	VALUEREP	REPORT SHOWING VALUE OF INVENTORY	
Press F8 to select this item.			

Now, take a look at the files used in the application. You want to change the name of the file used in the application. Move the cursor to the Files button field in the Builder Application section. (It indicates that the application uses one file.) Press F8 to open the Files Used window.

VISION:Workbench for DOS		Working:EXAMPLE	
COPY MOVE SEARCH REPLACE			
Application Name:EXAMPLE			RC4100
File description	Alternate	A I C D	U D C
Qual Name DD-Name Lbl	Key(s)	r C r B	User I/O Password/ p i h
0 ITEM M4OLD S		r F d I	Module Auth.-ID d r k
File or Array Name			

Here you see details of the file(s) used on the Files Used window.

To learn more about specific fields, ask for extended Help.

Put the cursor over ITEM and press F1.

VISION:Workbench for DOS				Working:EXAMPLE											
COPY				MOVE				SEARCH				REPLACE			
Application Name:EXAMPLE										RC4100					
File description		Alternate		A I C D		User I/O		Password/		U D C					
Qual Name	DD-Name	Lbl	Key(s)	r	C	r	B	Module	Auth.-ID	p	i	h			
0	ITEM	M40LD	S												
F1	F2	F3	F4	F5	F6	F7	F8	F9	Alt+F10						
Help	Names	Close	Menu Bar	Save	Choices	Validate	Open	Insert	Delete						

This displays the function key bar at the bottom of the window. Now press F1 again to see the help information for this field.

VISION:Workbench for DOS				Working:EXAMPLE														
COPY				MOVE				SEARCH				REPLACE						
Application Name:EXAMPLE										RC4100								
File description		Alternate		A I C D		User I/O		Password/		U D C								
Qual Name	DD-Name	Lbl	Key(s)	r	C	r	B	Module	Auth.-ID	p	i	h						
0	ITEM	M40LD	S															
Help Information MORE : ↓ The name which identifies this File. Enter one of the following: <table border="0"> <tr> <td>Type of File</td> <td>Entry</td> </tr> <tr> <td>-----</td> <td>-----</td> </tr> <tr> <td>Coordinated File, or Master File, or Transaction File, or Working Storage,</td> <td>File Name as it appears in the File Definition.</td> </tr> </table>													Type of File	Entry	-----	-----	Coordinated File, or Master File, or Transaction File, or Working Storage,	File Name as it appears in the File Definition.
Type of File	Entry																	
-----	-----																	
Coordinated File, or Master File, or Transaction File, or Working Storage,	File Name as it appears in the File Definition.																	

The text in the pop-up window explains the entry. Press PgDn (or ↓) and PgUp (or ↑) to scroll through the Help window. Press F3 to close the Help Information window.

VISION:Workbench for DOS				Working:EXAMPLE			
COPY MOVE SEARCH REPLACE							
Application Name:EXAMPLE						RC4100	
				A I C D		U D C	
File description				r C r B		Module	Auth.-ID
Qual Name	DD-Name	Lbl	Key(s)	r F d I			
0	ITEMAST	M4OLD	S				
File or Array Name							

Change the file name from ITEM to ITEMAST (to correspond with your file definition change). Press F3 to close the window and return to the application level.

VISION:Workbench for DOS				Working:EXAMPLE			
X=EXPORT RETRIEVE							
Application Name:EXAMPLE						RC4000	
Builder Application Sections							
Files.....:	1	Control	Params	Transact:	0	Comments:	No
Report Mgr:	No	Procs....:	1	Requests:	0	Reports.:	1
						Subfiles:	0
Type	Name	Description					
PROC	VALUE	;EACH MASTER-FILE RECORD CONTROL PROCEDU					
REPORT	VALUEREP	REPORT SHOWING VALUE OF INVENTORY					
Name of item; press F8 to select this item.							

Now, change the procedure. (The first line of the procedure is shown as description.) Move your cursor to VALUE and press F8.

VISION:Workbench for DOS		Working:EXAMPLE
COPY MOVE SEARCH REPLACE		
Procedure Name:VALUE	Type : N	RQ0100
Pre-Selection : N	Re-init Temps:	Maximum Items Selected:
<pre> ;EACH MASTER-FILE RECORD CONTROL PROCEDURE ; STOCKVAL: FIELD P 5 DEC 2 IF QANONHND > 0 LET T:STOCKVAL = QANONHND * ITMCOST CALL REPORT VALUERE END </pre>		
Enter a Procedure Statement		

Here is the procedure on the Procedure Definition window.

Move your cursor to ITMCOST and change it to UNITCOST.

VISION:Workbench for DOS		Working:EXAMPLE
COPY MOVE SEARCH REPLACE		
Procedure Name:VALUE	Type : N	RQ0100
Pre-Selection : N	Re-init Temps:	Maximum Items Selected:
<pre> ;EACH MASTER-FILE RECORD CONTROL PROCEDURE ; STOCKVAL: FIELD P 5 DEC 2 IF QANONHND > 0 LET T:STOCKVAL = QANONHND * ITMCOST LET T:STOCKVAL = QANONHND * UNITCOST CALL REPORT VALUERE END </pre>		
Enter a Procedure Statement		

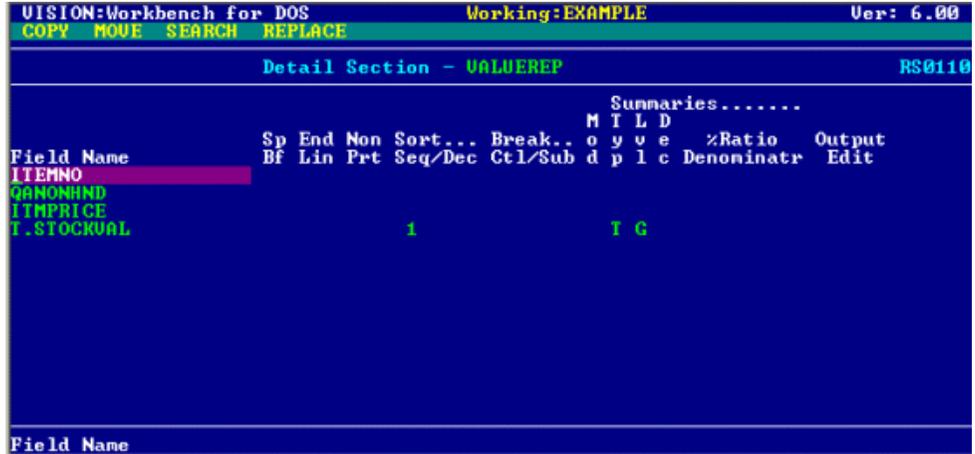
Press F3 to close this window and return to the Application Definition window.

VISION:Workbench for DOS		Working:EXAMPLE	
X=EXPORT RETRIEVE			
Application Name:EXAMPLE			RC4000
Builder Application Sections			
Files.....: 1	Control	Params	Transact: 0
Report Mgr:No	Procs...: 1	Requests: 0	Comments: No
		Reports.: 1	Subfiles: 0
Type	Name	Description	
PROC	VALUE	;EACH MASTER-FILE RECORD CONTROL PROCEDU	
REPORT	VALUEREP	REPORT SHOWING VALUE OF INVENTORY	
Name of item: press F8 to select this item.			

Now, look at the report definition. Move the cursor to VALUEREP and press F8.

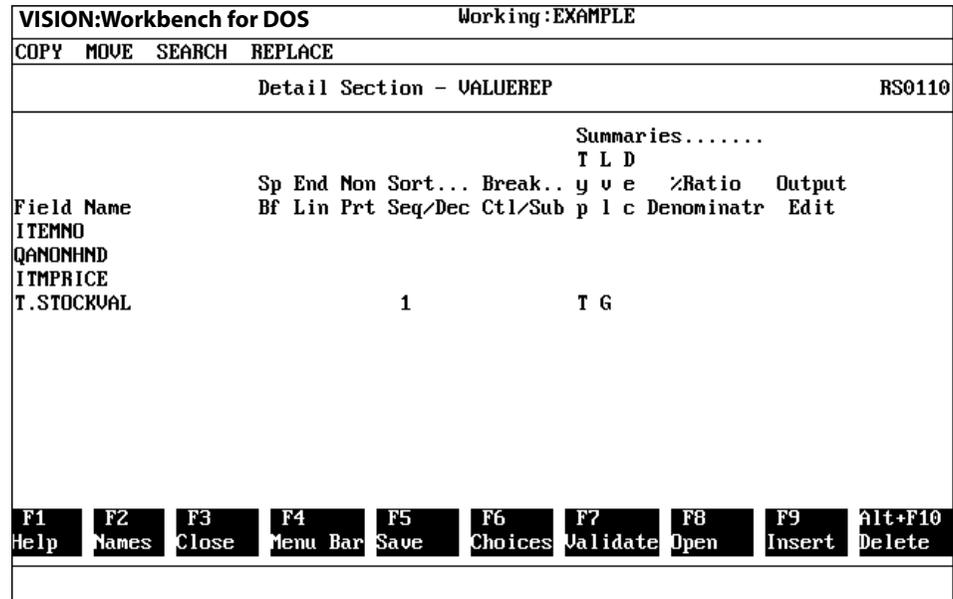
VISION:Workbench for DOS		Working:EX1		Ver: 6.00	
SIMULATE					
Report Name:VALUEREP				RS0100	
Report Layout Sections					
Preface: No		Detail: Yes		Columns: No	
Title: Yes		Xtitle: No		Summary: No	
Comments: Yes					
Physical Page			Other		
Vertical Spacing.:			Alt Report File:		
Line Numbers.....:			Abbrev. Format.:		
			Output Method...:No		
			Summary File...:No		
Logical Page			Titles		
Max Lines / Page.:			First Page Number...:		
Max Pages.....:			Page Number Position:		
Width of Page....:80			Single Title?.....:		
Height of Page...:			Repeating Subtitles.:		
Number across....:			Title at Bottom?....:		
			Date Position.....:		
			Report Date...: TODAY		
Required Name					

You are now looking at the application report layout on the Report Definitions window. To see the fields used in this report, position your cursor over Detail (Yes indicates that field specifications exist). Press F8 to display them.



The report contents are listed on the Report Details window.

You want to insert the new field name. To find out how, ask for the function key bar by pressing F1.



Note that the F9 function key performs an Insert. Position the cursor on QANONHND and press F9.

A blank line is inserted for you.

VISION:Workbench for DOS		Working:EXAMPLE	
COPY	MOVE	SEARCH	REPLACE
Detail Section - VALUEREP			RS0110
Summaries.....			
T L D			
Field Name	Sp End Mon Sort... Break..	y v e	%Ratio Output
ITEMNO	Bf Lin Prt Seq/Dec Ctl/Sub	p l c	Denominatr Edit
QANONHND			
ITMPRICE			
T.STOCKVAL	1		T G
Field Name			

Type in the new field name UNITCOST. Press F3 to return to the previous window.



To change the report title, use the Tab or cursor keys to move to the Title button field (Yes indicates that a title already exists). Press F8.

VISION:Workbench for DOS		Working:EXAMPLE	
CENTER	EDIT		
Line	1	Col	1
		Report Title - VALUEREP	RS0130
REPORT TITLE - VALUEREP REPORT FOR ABC COMPANY REPORT OF THOSE ITEMS THAT ARE IN STOCK AND THE VALUE OF THE INVENTORY			

Here is the current title on the Report Titles window.

Type over the first line of the old title with the following:

REPORT TITLE - UNIT COST ITEM VALUE REPORT

Note: Here, you can enter uppercase and/or lowercase entries. Simply use the SHIFT or CAPS LOCK to enter uppercase characters.

The title lines on this window are centered. Press Enter to see your new line centered with the others.

VISION:Workbench for DOS		Working:EXAMPLE	
CENTER EDIT			
Line	1	Col	1
Report Title - VALUEREP			RS0130
REPORT TITLE - UNIT COST ITEM VALUE REPORT REPORT FOR ABC COMPANY REPORT OF THOSE ITEMS THAT ARE IN STOCK AND THE VALUE OF THE INVENTORY			

Here is the title with the change.

Press F3 to return to the previous window.

The screenshot shows the 'SIMULATE' screen in VISION:Workbench for DOS. At the top, it displays 'VISION:Workbench for DOS', 'Working:EX1', and 'Ver: 6.00'. Below this is a green bar with 'SIMULATE' in yellow. The main area is blue with white text. At the top right, it says 'Report Name: VALUEREP' and 'RS0100'. The screen is divided into several sections: 'Report Layout Sections' with options for Preface, Detail, Columns, Comments, Title, Xtitle, and Summary; 'Physical Page' with options for Vertical Spacing and Line Numbers; 'Logical Page' with options for Max Lines / Page, Max Pages, Width of Page (set to 80), Height of Page, and Number across; 'Other' with options for Alt Report File, Abbrev. Format, Output Method, and Summary File; and 'Titles' with options for First Page Number, Page Number Position, Single Title?, Repeating Subtitles, Title at Bottom?, Date Position, and Report Date (set to TODAY). At the bottom left, it says 'Required Name'.

Press F3 again to return to the application level.

VISION:Workbench for DOS		Working:EXAMPLE										
X=EXPORT RETRIEVE												
Application Name:EXAMPLE			RC4000									
Builder Application Sections												
Files.....:	1	Control	Params									
Report Mgr:No	Procs...:	1	Requests: 0									
			Transact: 0									
			Comments: No									
			Reports.: 1									
			Subfiles: 0									
<table border="1"> <thead> <tr> <th>Type</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>PROC</td> <td>VALUE</td> <td>;EACH MASTER-FILE RECORD CONTROL PROCEDU</td> </tr> <tr> <td>REPORT</td> <td>VALUEREP</td> <td>REPORT SHOWING VALUE OF INVENTORY</td> </tr> </tbody> </table>				Type	Name	Description	PROC	VALUE	;EACH MASTER-FILE RECORD CONTROL PROCEDU	REPORT	VALUEREP	REPORT SHOWING VALUE OF INVENTORY
Type	Name	Description										
PROC	VALUE	;EACH MASTER-FILE RECORD CONTROL PROCEDU										
REPORT	VALUEREP	REPORT SHOWING VALUE OF INVENTORY										
Name of item; press F8 to select this item.												

Before you look at a simulated version of the report, you should validate the application. Press F7 to validate the application.

VISION:Workbench for DOS		Working:EXAMPLE										
X=EXPORT RETRIEVE												
Application Name:EXAMPLE			RC4000									
Builder Application Sections												
Files.....:	1	Control	Params									
Report Mgr:No	Procs...:	1	Requests: 0									
			Transact: 0									
			Comments: No									
			Reports.: 1									
			Subfiles: 0									
<table border="1"> <thead> <tr> <th>Type</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>PROC</td> <td>VALUE</td> <td>;EACH MASTER-FILE RECORD CONTROL PROCEDU</td> </tr> <tr> <td>REPORT</td> <td>VALUEREP</td> <td>REPORT SHOWING VALUE OF INVENTORY</td> </tr> </tbody> </table>				Type	Name	Description	PROC	VALUE	;EACH MASTER-FILE RECORD CONTROL PROCEDU	REPORT	VALUEREP	REPORT SHOWING VALUE OF INVENTORY
Type	Name	Description										
PROC	VALUE	;EACH MASTER-FILE RECORD CONTROL PROCEDU										
REPORT	VALUEREP	REPORT SHOWING VALUE OF INVENTORY										
Validation complete - no errors - Press any key.												

After validating the application, press any key to continue.

VISION:Workbench for DOS		Working:EXAMPLE
X=EXPORT RETRIEVE		
Application Name:EXAMPLE		RC4000
Builder Application Sections		
Files.....: 1	Control	Params
Report Mgr:No	Procs...: 1	Requests: 0
Transact: 0	Reports.: 1	Comments: No
Subfiles: 0		
Type	Name	Description
PROC	VALUE	:EACH MASTER-FILE RECORD CONTROL PROCEDU
REPORT	VALUEREP	REPORT SHOWING VALUE OF INVENTORY

Name of item; press F8 to select this item.

Now that the application has been validated, see what a simulated sample of the report looks like.

Leave the cursor on VALUEREP and press F8.

VISION:Workbench for DOS		Working:EX1	Ver: 6.00
SIMULATE			
Report Name:VALUEREP		RS0100	
Report Layout Sections			
Preface: No	Detail : Yes	Columns: No	Comments: Yes
Title : Yes	Xtitle : No	Summary: No	
Physical Page		Other	
Vertical Spacing.:		Alt Report File:	
Line Numbers.....:		Abbrev. Format.:	
		Output Method...:No	
		Summary File...:No	
Logical Page		Titles	
Max Lines / Page.:		First Page Number...:	
Max Pages.....:		Page Number Position:	
Width of Page....:80		Single Title?.....:	
Height of Page...:		Repeating Subtitles.:	
Number across....:		Title at Bottom?....:	
		Date Position.....:	
		Report Date...: TODAY	
Required Name			

Press F4 to move the cursor to the SIMULATE menu bar item and press Enter.

VISION:Workbench for DOS		Working:EXAMPLE				
Line	2	Col	1	Report Simulation - VALUEREP	RS0300	
01/15/95		REORT TITLE - UNIT COST ITEM VALUE REPORT REPORT FOR ABC COMPANY REPORT OF THOSE ITEMS THAT ARE IN STOCK AND THE VALUE OF THE INVENTORY			PAGE 1	
		ITEM NUMBER	ITEM COST	QUANTITY ON HAND	ITEM PRICE	STOCKVAL
		AAAAAAA	\$.01	1	\$.01	.01
		ZZZZZZZ	9,999.99	99,999	9,999.99	9,999,999.99
GRAND	TOTAL					9,999,999.99

This is a sample of what the report would look like if it were printed. Notice the changed title. If you recall, when you changed the field name you did not change the column heading — it remains ITEM COST in the report.

Press F3 twice to return to the Application Definition window.

VISION:Workbench for DOS		Working:EXAMPLE			
<input checked="" type="checkbox"/> EXPORT		RETRIEVE			
Application Name:EXAMPLE					RC4000
Builder Application Sections					
Files.....:	1	Control	Params	Transact: 0	Comments: No
Report Mgr:No		Procs....:	1	Requests: 0	Reports.: 1
				Subfiles: 0	
Type	Name	Description			
PROC	VALUE	;EACH MASTER-FILE RECORD CONTROL PROCEDU			
REPORT	VALUEREP	REPORT SHOWING VALUE OF INVENTORY			
Export the current application					

Currently, the application is in an internal format only readable by VISION:Workbench. To process the application by VISION:Builder, this internal format must first be converted to a format readable by VISION:Builder. This conversion is done by using EXPORT.

EXPORT creates 80-position images that you can add job control to and send to the host for processing by VISION:Builder as M4INPUT.

Press F4 to move the cursor to the menu bar. Select EXPORT.

VISION:Workbench for DOS		Working:EXAMPLE		
Application Name:EXAMPLE		RC4000		
- Builder Application Sections -				
Files.....: 1	Control	Params	Transact: 0	Comments: No
Report Mgr:No	Procs...: 1	Requests: 0	Reports.: 1	Subfiles: 0
Type	Name	Description		
PROC	VALUE		PROCE	
REPORT	VALUEREP			
		<div style="border: 1px solid black; padding: 5px;">Select Mainframe Processing: Normal Execution Sample Reports Only Syntax Validation Only</div>		
Press ENTER or F8 to select this item				

The files are given the name of the application with a suffix appended to distinguish their contents. One of the files has .DEF appended; this file contains definition statements for the application. The other file has .RUN appended; this file contains procedure and report statements. The files generated by this application are named EXAMPLE.RUN and EXAMPLE.DEF. You can use your PC-host communication program to transfer these files to the host for processing.

VISION:Workbench for DOS		Working:EXAMPLE	
X=EXPORT RETRIEVE			
Application Name: EXAMPLE			RC4000
Builder Application Sections			
Files.....:	1	Control	Params
Report Mgr:No		Procs....:	1
		Requests:	0
		Transact:	0
		Comments:	No
		Reports.:	1
		Subfiles:	0
Type	Name	Description	
PROC	VALUE	;EACH MASTER-FILE RECORD CONTROL PROCEDU	
REPORT	VALUEREP	REPORT SHOWING VALUE OF INVENTORY	
Required Name			

Note: EXPORT creates the 80-position statements, but it does not save the application in its PC form.

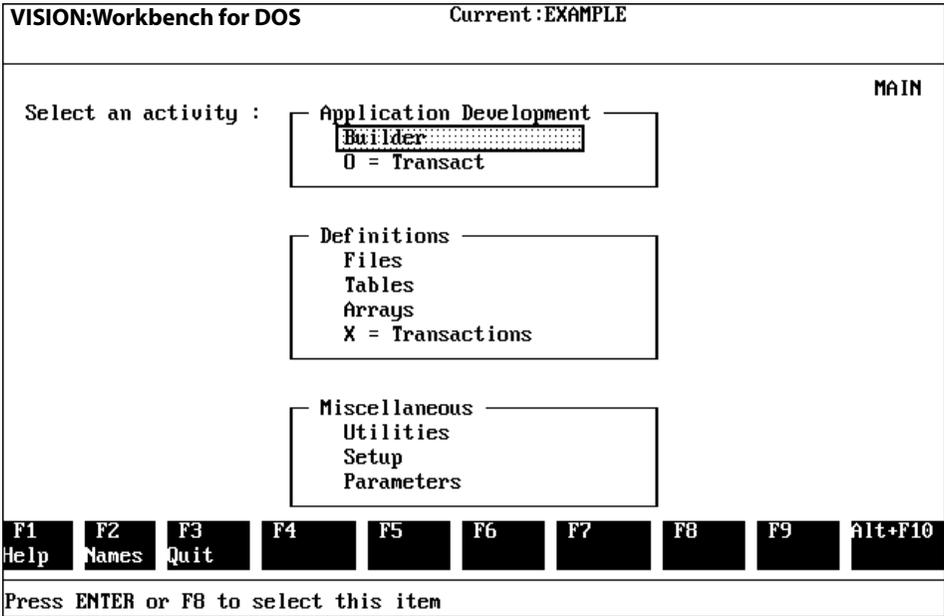
Save the application on your PC. Press F5. (You can use F1 if you do not remember which key saves the application.) Press F3 to return to the Application List window.

VISION:Workbench for DOS		Working:EXAMPLE		
USERINFO		User file index information		FI0100
Type	Name	Location		
BUILDER	EXAMPLE	Currently in memory		
BUILDER	EX1	C:\MPCXX\SAMPAPPS		
BUILDER	OLYMPICS	C:\MPCXX\SAMPAPPS		
BUILDER	TRANRUN	C:\MPCXX\SAMPAPPS		

Name of item: press F8 to select this item.

Notice that the new application has been added to the list of applications.

Press F3 to return to the Main Menu.



The Builder application is complete.

Press F3 again to exit VISION:Workbench.

Creating VISION:Transact Applications

A VISION:Transact application consists of a series of screen interactions. During an interaction, a screen displays and the operator enters data from the terminal. Upon pressing the Enter key (or a PF key), the databases are synchronized with the data on the screen, records that correspond to the data on the screen are retrieved from the databases, appropriate updates are made to the databases, and, at the end of the interaction, a new screen is presented. At various points during the interaction cycle, user procedures can be activated to handle specialized processing.

At the highest level, a VISION:Transact application consists of a series of screens that display (and, subsequent to data entry, the display of a series of follow-on screens). This is where you start the design of a VISION:Transact application. First, you specify the screens and the follow-on screens. You do this by drawing a hierarchical picture where each box depicts a screen and the “children” (or dependents) of the box depict the appropriate follow-on screens. [Figure 9-1](#) illustrates this.

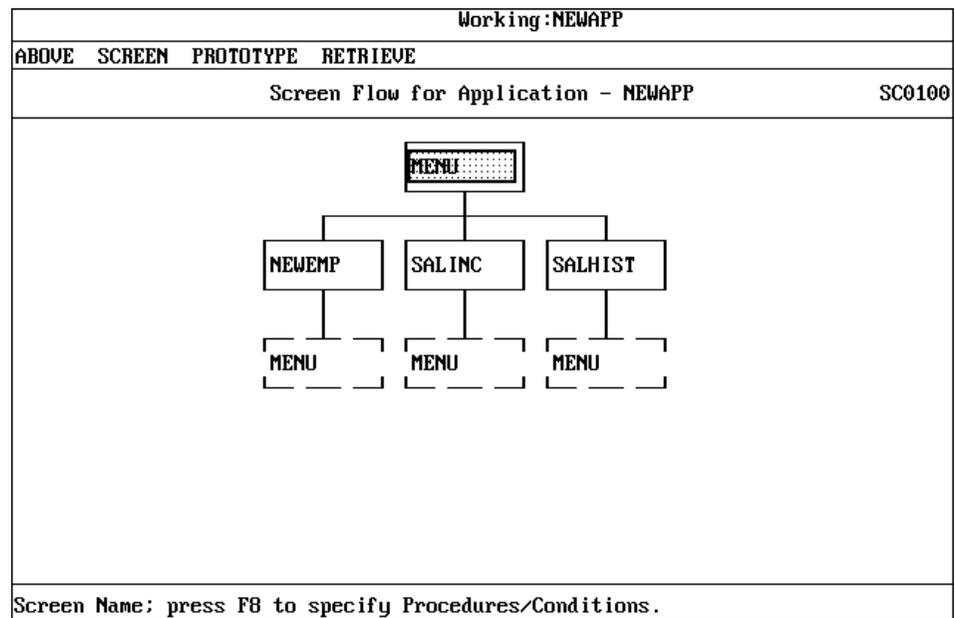


Figure 9-1 Screen Flow

Here you see the screen flow of the application all laid out. It is the same application that you will use in the walk-through in [Chapter 11, Walk-Through for VISION:Transact Applications](#). The root screen is the screen labeled MENU. This screen may have three different children screens, depending upon the function keys activated. The possible follow-on screens are NEWEMP, SALINC, and SALHIST. Each of these follow-on screens themselves can have a follow-on screen. In the illustration shown, this single follow-on screen is MENU (that is, each subordinate screen returns back to the main menu). Rather than having lines all over the screen which can become very misleading, the convention in VISION:Workbench is if the follow-on screen is a screen at a higher level, it is depicted with broken lines. You do not have to worry about this. It is taken care of automatically by VISION:Workbench. When you see a broken box, you know that this leg of the application is going to continue to a screen that displays at some other level of the hierarchy.

This simple structure chart defines the structure of the application. It is now appropriate to examine each of the interactions in detail. Do this by providing all the information relevant to a screen at the same time. Position the cursor on the box or screen for which you want to provide information and open a window. Press F8 to open the window. [Figure 9-2](#) illustrates the window that has been opened for the screen SALHIST.

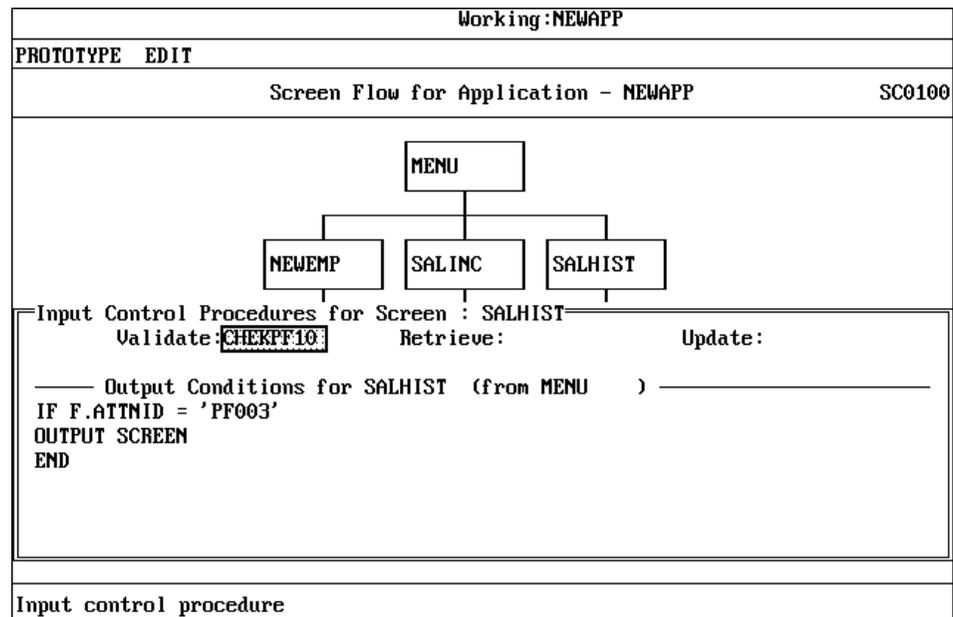


Figure 9-2 Conditions Window

This window consists of two parts. The top part provides the control procedures that are to be activated during the interaction triggered by the screen. The bottom part contains the conditions that cause the screen SALHIST to display in the first place.

First the control procedures. During the interaction, there are three points at which user procedures can be activated. These are for screen validation shown in the window as “Validate” (corresponding to the VISION:Transact type 1 requests); “Retrieve” which is the point after which the databases have been accessed but not yet updated (corresponding to the VISION:Transact type 2 requests); and the “Update” point which is immediately after updating the data records in memory (corresponding to the VISION:Transact type M requests). VISION:Workbench uses the concept of event controlled procedures whereby only a single procedure can be coded. The procedure, however, can invoke as many subprocedures as needed. The practice enables you to construct disciplined, structured applications which are easier to debug and maintain. The top half of the Conditions window corresponds to the VISION:Transact process of providing validation and computational requests.

The bottom part of the screen provides the conditions that caused the screen SALHIST to be displayed from its parent screen (MENU). The conditions are displayed using a free-form construct. In this example, it simply shows that SALHIST will display if you press PF3 when the MENU screen (translation) is activated. It is important to understand that the condition must occur during processing of the transaction activated by the parent screen in order for the current screen to display.

This is the area where you would provide any specialized output procedures, such as the VISION:Transact type O request. Assume that you had an output procedure (called XXPQZ) during the menu interaction that is to be used to create the SALHIST display. This would be shown with the conditions as follows:

```
IF F.ATTNID = 'PF003'  
  OUTPUT SCREEN CALL XXPQZ  
END
```

Execution of XXPQZ is part of the MENU interaction (transaction processing), but the whole purpose of the procedure XXPQZ is to display SALHIST. Therefore, it is important to SALHIST and not to MENU. When you look at the window corresponding to SALHIST, you have all the information that is relevant to SALHIST. You know what will cause it to be displayed from its parent; you are able to see any special conditions; and you are able to see any special processing that occurs during the interaction that is triggered by SALHIST. If you were to decide to delete SALHIST from the application, it is sufficient to delete the box from the screen flow; there is no need to go to other boxes to make any changes.

Notice the high-level top down application design approach that you are using. So far, you have sketched out the interactions, identified what screens follow what other screens, established which procedures are to be activated during interactions (transactions), and identified the conditions that are necessary for follow-on screens to appear. You have not, as of this point, designed the screens or coded the procedures.

To complete the application, you need to do just that. The details of doing this are described later in this chapter and in [Chapter 10, Screen Layout, Contents, and Actions](#). However, the process is briefly described below.

Procedures are simple. Procedures are written using the ASL language, which is fully described in the ASL Reference Guide.

You perform screen layout by simply painting a screen. This can actually be activated either from the main application menu or from the screen flow that was discussed above (press F4 and select SCREEN). There are three components to a screen. These are layout, contents, and actions. "Layout" is achieved simply by painting the screen.

"Contents" identifies the data that is to be displayed. To achieve this, you position the cursor to an appropriate position on the screen and open up the window (press F8) to identify the data fields and appropriate editing characteristics.

Note: You can only position data on the screen by positioning the cursor and opening the Contents window (F8). VISION:Workbench will insert the appropriate number of underscores. If you insert underscores, they are treated just like text.

"Actions" identifies the particular transaction activity that is to take place as a result of data entry or pressing a function key. The actions are identified on the same window that is used to identify the contents. This is fully discussed in [Chapter 10, Screen Layout, Contents, and Actions](#).

All of the above defines a complete interaction with respect to activity relevant to the screens. There are only two more components required to complete the application. These are to identify how the various databases are synchronized and to provide characteristics concerning the host execution environment, such as CICS or IMS™/DC. The specifications required for each of these application areas are provided by activating windows from the main application development screen. When you select the button field FILES, a window displays enabling you to state which files are active during the application and how they are synchronized with each other. When you activate the button field ENVIRONMENT, a window displays enabling you to provide the various CICS or IMS/DC characteristics.

In summary, an application fundamentally consists of only four components.

- Screens
- Files
- Procedures
- Environment

The steps for designing an application are:

- Define the screen flow.
- Define the screen conditions and identify control procedures.
- Define the screen layouts, contents, and actions.
- Write the procedures.
- Establish the file synchronization rules.
- Provide the information about the environment.

Even though this chapter has described how you can build an application in a logical straight-forward manner, there is no constraint upon the order in which you perform these tasks. Sometimes it is more convenient to define the file synchronization as the first step. It is your choice. An application really consists of the components listed above. While you can construct these components in any order, all components must be constructed before the application is complete.

Free-Form Language

Free-form logic statements are used in several places in the design of a VISION:Transact application. Free-form statements are used for writing procedures (Procs). In this case, they are collectively referred to as ASL (Advanced Syntax Language). Free-form statements are also used to specify the screen output conditions, field validations, and actions.

Technically speaking, ASL statements consist of statements that are common to more than one Computer Associates product. These statements are fully documented in the ASL Reference Guide. However, free-form statements that are unique to VISION:Workbench are documented in this guide. For ease of use, the syntax of the special commands required for VISION:Transact applications has been kept similar (and, in some cases, identical) to ASL. If you are writing Procs, you should see the *ASL Reference Guide*; if you are defining screen conditions, field validations, or actions, you should refer to this guide. For convenience, the following sections provide a complete summary of all the free-form statements used in the development of Transact applications.

Summary of Validation Statements and Output Conditions

Validation Statements

action	Identifies the actions to apply to the destination fields. The following terms are valid: DELETE, INSERT, UPDATE, UPDATE OR INSERT, REPEAT, IGNORE, ADD, SUBTRACT, REPLACE, REPLACE IF NONBLANK, CLEAR, and NOACTION.
CASE	Begins a group of statements within a CASE block that are to be performed when the CASE condition is true.
DO	Begins the DO block and identifies the destination field to which the actions apply.
ELSE	Precedes the default action when specifying dynamic actions.
END	Terminates a DO block.

- | | |
|--------|--|
| ON | <ol style="list-style-type: none">1. Specifies the action to be taken in the event of certain error conditions.2. Indicates the screen fields to be highlighted in the event of an error. |
| VERIFY | Specifies field level validation. |

Output Conditions

- | | |
|---------|---|
| OUTPUT | Identifies options that apply to the output screen. |
| RELEASE | Releases any screen synchronized database structures from the intermediate structure. |

Application Development Windows

This section describes the use of the VISION:Transact application development component of the VISION:Workbench. It is used to define the files, screen flow, structure and layout, procedures, and requests used by your VISION:Transact application. (Screen layout is discussed separately in [Chapter 10, Screen Layout, Contents, and Actions](#).) You enter this part of VISION:Workbench by selecting Transact under the Application Development heading on the Main Menu. [Figure 9-3](#) shows the structure of how the windows within the VISION:Transact application development component are organized.

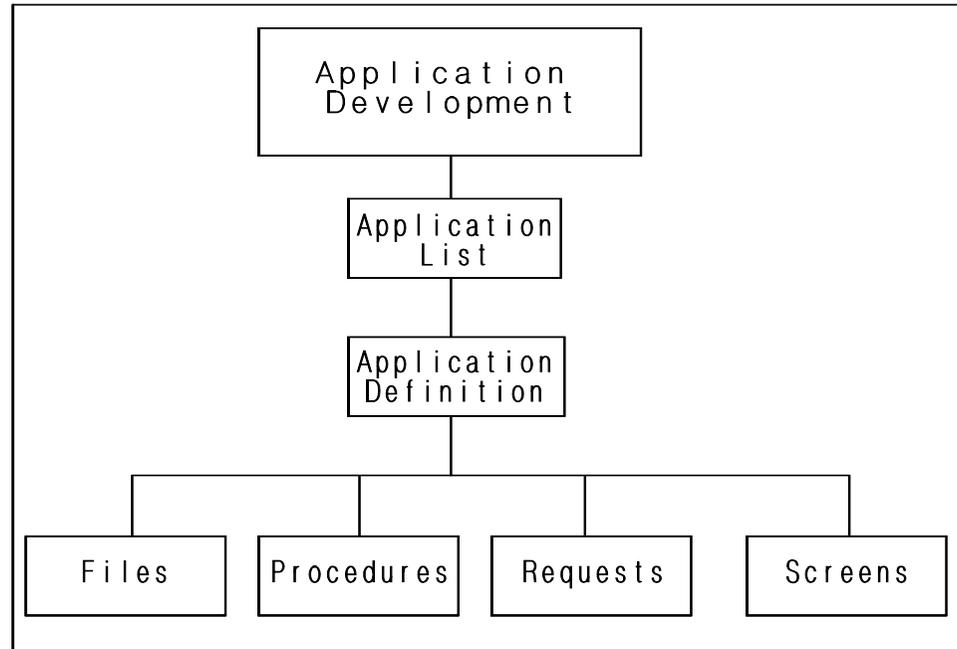


Figure 9-3 VISION:Transact Application Development Component

VISION:Transact Input Control Procedures

VISION:Workbench uses control procedures to help you structure your VISION:Transact application. Control procedures are based on events, such as screen interactions, and correspond to traditional VISION:Transact request types.

A VISION:Transact application is considered to be a fixed cycle of events. When a certain event occurs, a specific event controlled procedure is activated. Associated with each input screen, there are three particular events where your procedures may be expected:

- After screen input (corresponding to a VISION:Transact type 1 request).
- After the data is retrieved (that is, after screen synchronization has occurred — corresponding to a VISION:Transact type 2 request).
- After the data is updated (that is, after the data from the screen is applied to the records retrieved — corresponding to a VISION:Transact type M request).

Event controlled procedures are written in ASL. Only one event controlled procedure can be provided for each event in an interaction, but a procedure can call other procedures or conventional fixed format requests in any order.

VISION:Transact Processing Objects

VISION:Workbench for DOS applications are constructed by assembling together different objects. The types of objects that exist for VISION:Transact applications are:

- PROC A procedure written using ASL.
- REQUEST A procedure written using fixed format syntax.
- SCREEN A screen definition.

You create application objects by specifying the type of object you want to create and by completing the appropriate follow-on windows that open.

VISION:Transact Application List

The VISION:Transact Application List, shown in [Figure 9-4](#), is the first window opened after the Main Menu. This window lists the names of all existing VISION:Transact applications.

Current :		
USERINFO	User file index information	FI0100
Type	Name	Location
TRANSACT	EXAMPLE	C:\MPCXX\SAMPAPPS
TRANSACT	M501	C:\MPCXX\SAMPAPPS
TRANSACT	M502	C:\MPCXX\SAMPAPPS
TRANSACT	M503	C:\MPCXX\SAMPAPPS
TRANSACT	M504	C:\MPCXX\SAMPAPPS
TRANSACT	M508	C:\MPCXX\SAMPAPPS
TRANSACT	M509	C:\MPCXX\SAMPAPPS
TRANSACT	M510	C:\MPCXX\SAMPAPPS
Name of item: press F8 to select this item.		

Figure 9-4 VISION:Transact Application List Window

You select an existing application to be modified by placing your cursor on it and pressing F8. You can add new application names to the list at any time. You can add a new name to the end of the list or insert it at a specific location within the list. When you add a name to the list or select an existing one, the Application Definition window opens.

Application Definition Window

The Application Definition window is used to specify the various parts of the application. From the selections made on the Application Definition window, some or all of the following windows open and are used to complete the application definition:

- Files Used
- Screen Flow
- Environment
- Comments
- Screens
- Procedures (Procs)
- Requests

The Application Definition window, shown completed in [Figure 9-5](#), is divided into two parts. The top half of the window summarizes the various sections of the application. The first line consists of button fields. The selection of these fields takes you to another window. In some cases, the button field itself actually contains information about the underlying window. You open the window by placing the cursor over the button field and pressing F8. A window appropriate to the button field opens.

Working:EXAMPLE			
RETRIEVE X=EXPORT PROTOTYPE			
Application Name			RC5000
EXAMPLE			
Transact Application Sections			
Files	: Yes	Screen Flow:	Yes
Environment:	No	Comments:	No
Screens	: 4	Procs	: 3
Requests	: 0		
Type	Name	Share	Description
SCREEN	MENU	NO	INITIAL MENU SCREEN
SCREEN	NEWEMP	NO	ENTER EMPLOYEE DATA
SCREEN	SALINC	NO	PROVIDE SALARY INCREASE DATA
SCREEN	SALHIST	NO	DISPLAY THE SALARY HISTORY
PROC	CHEKPF	NO	CHECK THAT ONLY PF1, PF2 OR PF3 PRESSED
PROC	CHEKPF10	NO	CHECK THAT ONLY PF10 PRESSED
PROC	SALCALC	NO	CALCULATE SALARIES FROM INCREASE DATA
Required Name			

Figure 9-5 Completed Application Definition Window

The second line of the window is a summary of various objects defined within the system. A count of procedures, requests, and screens is provided. This section of the window cannot be modified and does not operate as button fields.

The bottom portion of the window is used as a list to describe the type of application object and to assign it a name. The valid application types are Screens, Procs, and Requests.

The Share column specifies whether or not the item can be shared among applications. This entry is applicable when the application is saved at the application level. All items that are saved at the item level can be shared.

Under Description, you can document the item. When you select an existing entry from the list or make a new one and select the appropriate menu bar item, the appropriate definition windows open.

Application Definition Menu Bar Items

The Application Definition menu bar displays on the Application Definition window. Press F4 to move the cursor to the menu bar area. The following menu items are available from this window:

RETRIEVE	Retrieves application items from the library into the current application.
EXPORT	Creates PC-DOS files of your application to be sent to the host.
PROTOTYPE	Displays the root level screen, and allows you to step through the flow of screens, screen by screen.

Using RETRIEVE

When you select the RETRIEVE menu item, the menu bar items PROCEDURE, REQUEST, and SCREEN display. Select the type of item you want to retrieve; a list of available items of that type displays. Select the application object you want and press Enter. The selected item is inserted on the Application Definition window where you positioned your cursor. All related windows are included in the application.

Using EXPORT

The EXPORT menu item puts your application in a format that can be sent to the host. See [Chapter 12, Utilities and Host Activity](#) for information on this subject.

Using Prototype

Using VISION:Workbench for DOS, you can prototype your application. Prototyping consists of displaying a screen from the application and choosing a follow-on screen. Prototyping is invoked from the Application Definition window or the Screen Flow window. You can prototype only an application that has been successfully validated.

Screens are displayed using a full simulation of the 3270 representation. Because of the existence of different terminal types, this may be a simulation of a screen that is larger than the PC screen. You can move the PC screen left, right, up, and down, by using the arrow keys. Keyboard handling emulates 3270 usage. For example, Tab, BackTab, PgUp, and PgDn only move to unprotected data fields. You can also type sample information into unprotected data fields (except those with DARK intensity).

The screen display differs from the format used in the Screen Layout in several ways:

- Field designators do not display.
- The default cursor position does not display.
- Output fields (except those with DARK intensity) are filled in with sample data. Xs are used for character fields, and 9s are used for numeric fields. Date flag fields contain data appropriate to their usage.

You can select follow-on screens in one of two ways:

- Press Escape to display a list of available follow-on screens. By positioning the cursor and pressing Enter, you can select which of the follow-on screens you want to see. Press Escape or F3 in response to this list to terminate prototyping. You also terminate prototyping if you press Escape in response to an application screen that has no follow-on screen.
- If the Screen Control conditions consist of simple logical relations in the form of F.ATTNID = `PFnnn' or F.ATTNID = `ENTER', VISION:Workbench responds to a user-pressed function key or Enter key by choosing the corresponding output screen.

Files Used Windows

You enter the Files Used information in two separate windows. To open the first window, you select the Files button field on the Application Definition window. The first window gives an overview of the file types used. Each of the file entries is a button field with the variable portion indicating the number of files of that type currently defined.

When you select any of the file types in the first window, a second window displays containing the necessary information for that file type. In this case, the secondary window displays together with the primary window, as shown in [Figure 9-6](#).

In each of the available secondary windows, you provide the file name for the file being assigned to that class of file as well as a qualifier and any other processing characteristics. [Figure 9-6](#) shows the file information windows for databases.

Working:EXAMPLE							
Files for Application : EXAMPLE							RC5100
Data Bases.....:	1	Pseudo Files :	0				
Working Storage :	0	PCBs.....:	0				
		Screen		Dynamic			
		--- Sync D.B. ---		----- Sync D.B. -----			
File Name	Qual	Method	Path	Call	Field..1	Field..2	Length
PERSFILE	N	UPDATE					
Name of Data Base							

Figure 9-6 Files Used Windows

Screen Flow Window

The Screen Flow window, shown in [Figure 9-7](#), is opened as a result of selecting the Screen Flow button field on the Application Definition window. The screen flow of the application is completed by drawing a hierarchy of screens. Screens can be entered and edited from the screen flow diagram.

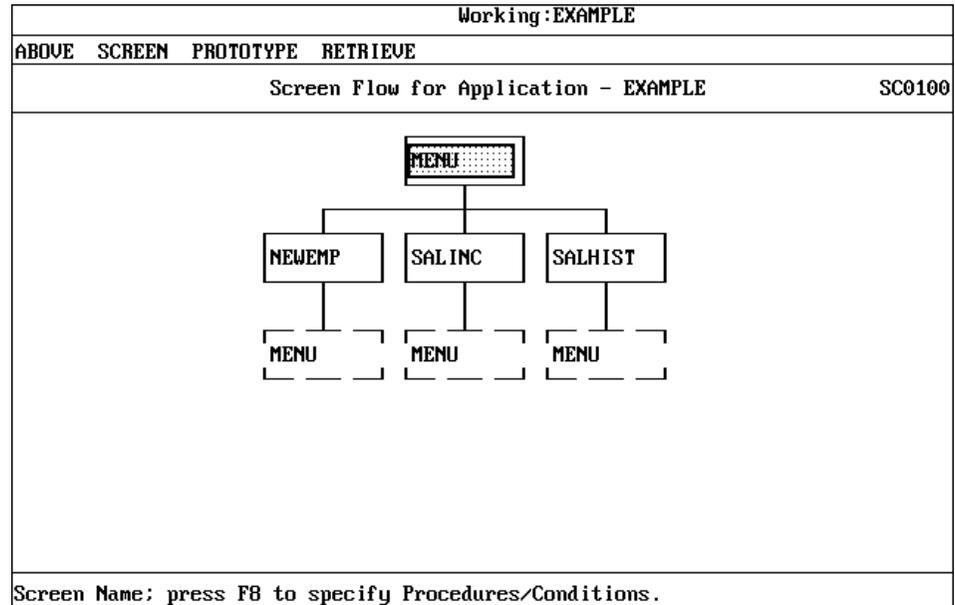


Figure 9-7 Screen Flow Window

The screen hierarchy is drawn in almost exactly the same way as the file definition hierarchy is drawn (see [Chapter 5, Entering Definitions](#)). Each of the boxes on the window represents an application screen. Everything that is associated with a particular screen can be obtained by positioning the cursor into the box that represents the screen and pressing F8. This opens the Conditions window for the particular screen. The Conditions window, which establishes the conditions under which one screen is displayed following another, is discussed in a later section.

There are a few differences between defining files and defining screen flow. First, when defining screen flow you can have multiple root boxes defined. Also, with screen flow, it is possible for the same screen to appear at different places in the flow (that is, to be displayed under different conditions). If a screen is represented on the screen flow diagram by more than one box, the second and subsequent boxes are displayed with a dashed outline. Although the Conditions window can be displayed for these dashed boxes, do not change or edit the top part of the Conditions window (the event controlled procedures). The event controlled procedures should only be changed from the screen representation with the solid box.

Each box on the screen flow represents a screen. The branches show the potential follow-on screens. A dashed box can be thought of as a connector as used in flowcharting (that is, it takes you back to another point in the hierarchy of screens).

VISION:Transact Screen Layout

Screen layouts are specified in a “what-you-see-is-what-you-get” manner. You type literal text on the display exactly as you want it to appear. Screen fields are entered into the layout not by typing field designator characters but by positioning the cursor to where you want the field to appear and pressing the appropriate function key.

Display attributes, such as highlighting and color, are specified by literally “painting” areas of the screen using the cursor control keys.

Multiple segment occurrences are specified using the COPY selection (from the EDIT menu) to copy a section of the layout one or more times.

These techniques are described in greater detail in [Chapter 10, Screen Layout, Contents, and Actions](#).

VISION:Transact Screen Definition

VISION:Workbench allows screens to be defined within the context of an application, unlike VISION:Transact which requires a separate definition step. In VISION:Workbench, applications can share screens. To activate the share feature, enter YES in the Share column of the VISION:Transact application and save the application.

Screen Flow Menu Bar Items

The Screen Flow menu bar is displayed on the Screen Flow window. You press F4 to move the cursor to the menu bar area and select the menu bar item you want by either moving the cursor to it and pressing Enter, or by entering its first letter (for example, S selects the SCREEN menu item). The following menu bar items are available during the screen flow editing process:

ABOVE	In conjunction with the F9 insert key, inserts a blank screen box above any box from which this item is selected.
SCREEN	Opens the Screen Definition window for the screen on which the cursor is currently positioned.
PROTOTYPE	Displays the root level screen and allows you to step through the flow of screens, screen by screen.
RETRIEVE	Retrieves screens from the VISION:Workbench library.

Using ABOVE

The ABOVE menu item allows you to modify the screen flow by inserting a new screen box above an existing screen box. It is selected on the Screen Flow window. You use ABOVE by positioning your cursor in the screen box that you want to insert a new screen box immediately above. After selecting ABOVE, press the F9 insert key.

A new blank box appears immediately above the box in which your cursor was positioned (see [Figure 9-8](#)); the cursor is repositioned in the new box. You can type a screen name into the new box, identifying the screen associated with the box.

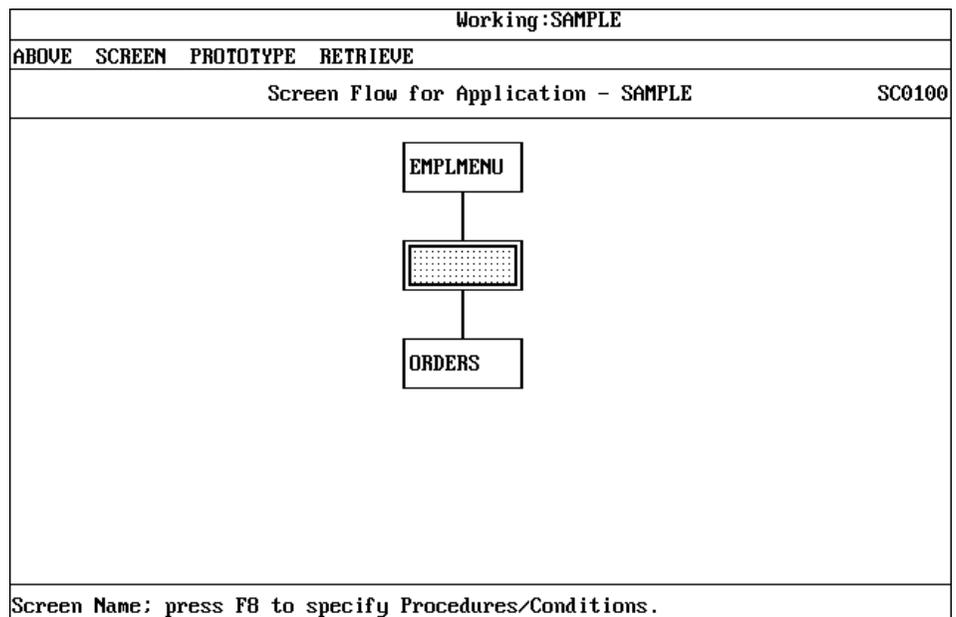


Figure 9-8 New Screen Box After Selecting ABOVE

Using SCREEN

The SCREEN menu bar item opens the Screen Structure Definition window which is shown in [Figure 9-9](#).

Working: EXAMPLE			
MOVE			
Layout : Yes		Comments: No	Screen Name SALHIST SD0100
Input Transaction Code:		BMS/MFS Name....:	
Device ID.....:		Field Designator:	
Process Empty Segments:			
Segment	Occurs	Display - Virtual	Fields.....
1	1	4	0
2	9	3	0
Required Name			

Figure 9-9 Screen Structure Definition Window

The top section of the window provides information pertaining to this screen and its environment. Layout and Comments are button fields. Selecting Layout opens the Screen Layout window where you can paint the screen. Screen painting is discussed in [Chapter 10, Screen Layout, Contents, and Actions](#). Selecting Comments opens the Comments window.

The lower section of the window allows you to define the structure of the screen. Nine levels of structure are provided with eight entries for the number of occurrences of each section. Level 1 can only have one occurrence; this is hard-coded and cannot be modified.

If you are creating a structured screen, enter the number of segment occurrences required for each of the sections of the structure.

The Display Fields and Virtual Fields data summarizes the number of displayable and virtual fields defined in that screen segment. These fields are button fields. Each of the follow-on windows that appear can be scrolled (with arrow keys) and zoomed (with Alt+F3). When you use the ZOOM key, the window expands to display more field names.

When you OPEN the Display Fields window, a list of the display fields in the segment is shown. From this list, displayable screen fields can be deleted, but other functions must be done in conjunction with the Layout display.

When you OPEN the Virtual Fields window, a virtual field can be defined by providing a name. It can be added to the end of the list or inserted into the list by use of the Insert key. Virtual field names can be inserted, deleted, and modified within this list.

When the name is provided the virtual field Contents/Action window can be opened. A virtual field must have a Contents window completed before the screen definition can be fully validated. The statements in the additional Validation/ Actions are identical to those for displayable fields and are discussed in [Chapter 10, Screen Layout, Contents, and Actions](#).

Using PROTOTYPE

The PROTOTYPE menu item can be invoked from either the Application Definition window or the Screen Flow window. Information on the use of PROTOTYPE can be found under Application Definition earlier in this section.

Using RETRIEVE

The RETRIEVE menu item allows you to retrieve screens (with a SHARE value of YES) from the VISION:Workbench library. VISION:Workbench allows you to save individual screens into a library for later use.

Conditions Window

To display the Conditions window for a given screen, position the cursor in the appropriate box in the screen flow diagram and press the OPEN key (F8).

The Conditions window identifies the input control procedures to be used in processing the screen's input as well as the conditions under which the screen is to be output as a follow-on screen from the parent screen. A Conditions window is shown in [Figure 9-10](#).

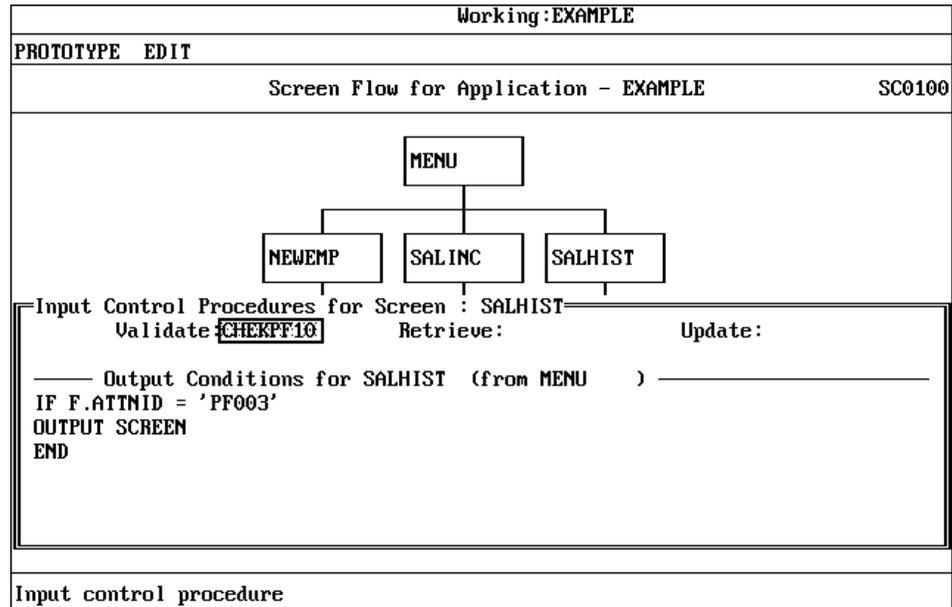


Figure 9-10 Conditions Window

The Input Control Procedures entries (Validate, Retrieve, and Update) identify the control procedures which handle any special input processing for the screen. These procedures correspond to VISION:Transact fixed format request types 1, 2, and M, respectively. Only one procedure can be specified for each of the events, but the procedure can call other procedures as subroutines.

The Output Conditions section contains one or more output statements that identify options that apply to the output screen. All parameters on the output statement are optional. Leaving the Output Conditions section empty is equivalent to specifying OUTPUT SCREEN. The bottom part of the Conditions window is scrollable to allow for lengthy, complex conditions. The window can be expanded to a full screen by means of the ZOOM key (Alt+F3). The ZOOM key is a toggle switch that expands/contracts a window to/from a full size window.

The format of the output statement is:

```

OUTPUT SCREEN DESTINATION (dest-name)
        CALL (procedure-name) OPTIONS (OC-statement-options)
        OUTLIM (output-limit)
    
```

The SCREEN keyword is optional and purely for readability.

Dest-name is a field or constant whose value is the name of the destination to which this output screen is to be sent. If multiple destinations are required, one OUTPUT statement must be specified for each destination. If no DESTINATION keyword is specified, output is to the primary terminal.

Procedure-name is the name of the procedure used to help build this output screen. This procedure corresponds to a VISION:Transact type O request.

OC-statement-options are the output control options used to supply control information that processes and routes this output screen. Any combination of the following options are valid:

NO_INDICATORS	Suppresses the indicators used for missing, invalid, numeric overflow, and null fields.
CONVERSATIONAL	In IMS/DC, inserts the SPA in the message queue before the first output message allowing the conversation to be continued by the next transaction. In CICS, enables the application to converse with the primary terminal.
START	Initiates a transaction for the transaction contained in the TRANCODE flag to run asynchronously and to pass one page of output to it.
RECYCLE	Outputs and re-inputs one page of the output screen.

Output-limit specifies the number of pages produced by an output screen. Valid values are 1 to 999.

The RELEASE statement functions like the VISION:Transact SG operator on an SC statement. It is used to release screen synchronized database segments from the intermediate structure and affects any OUTPUT statements that follow it. Its format is:

RELEASE segment-name(s)

Comments can be placed anywhere in the Output Conditions following a semicolon (;).

Conditions can be applied to the output screen by use of IF/END or DO CASE/END statements. The conditions are written and operate as described in the ASL Reference Guide; however, the ELSE statement cannot be used. If there is more than one screen defined as output from a parent screen, only one of the conditions windows can be written without IF or DO CASE statements.

Selecting a Follow-on Screen

The procedure for selecting a follow-on screen is as follows:

1. Each of the output conditions for each of the possible follow-on screens is evaluated (from left to right).
2. The first true condition causes all associated OUTPUT statements to be performed. Subsequent conditions and outputs are bypassed.
3. If no condition is true, all *unconditional* OUTPUT statements (for all of the follow-on screens) are performed.

In the event that two or more screens have identical output conditions, all such screens will be output. For example:

```

----Output Conditions for SCREENB (from SCREENA )-----
IF F.ATTNID EQ 'PF001'
    OUTPUT SCREEN
END

----Output Conditions for SCREENC (from SCREENA )-----
IF F.ATTNID EQ 'PF001'
    OUTPUT SCREEN DESTINATION 'TERMABC'
END
    
```

In this example, if F.ATTNID EQ `PF001', both SCREENB and SCREENC will be output.

Note that the same screen can be output to multiple destinations:

```

----Output Conditions for SCREENB (from SCREENA )-----
IF F.ATTNID EQ 'PF001'
    OUTPUT SCREEN
    OUTPUT SCREEN DESTINATION 'TERMXYZ'
END
    
```

Application Environment Window

The Application Environment window, shown in [Figure 9-11](#), contains host environment information. It is opened by selecting Environment on the Application Definition window.

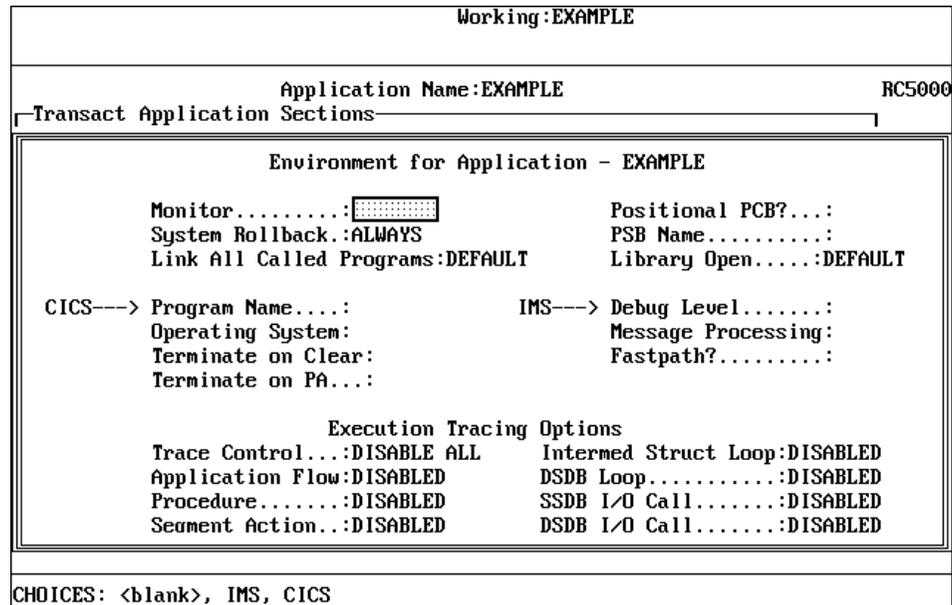


Figure 9-11 Application Environment Window

Application Comments Window

The Comments window is opened as a result of selecting the Comments button field on the Application Definition window. The Comments window is used to specify comments or descriptive information about the application. They can be entered in the window in any format.

Screen Definition Window

The Screen Definition window can be accessed in two ways. (VISION:Workbench windows that are used to define a screen and associated information are the same, regardless of how the screen is accessed.) The first way to access the Screen Definition window is through the Screen Flow menu bar item SCREEN. Refer to the Screen Flow window earlier in this chapter for information on this method.

The second way to access the Screen Definition window is from the Application Definition window. From the Application Definition window, you can open the Screen Definition window by either selecting an existing screen or entering a new screen. To enter a new screen, type an S in the Type entry (the rest of the word is filled in for you) or press F6 for Choices and make your selection. Enter the name of the screen, a description, and press F8.

Screens created using the second method (from the lower portion of the Application Definition window) become objects but are not connected into the application. They must be connected into the application using the screen flow. Screens are accessible in this manner primarily for the convenience of editing an existing screen without the necessity of stepping through the screen flow.

Refer to Using SCREEN earlier in this chapter for more information on how to use the Screen Definition window.

Procedure Definition Window

The Procedure Definition window is accessed from the Application Definition window. To open the Procedure Definition window, either select an existing procedure or enter a new one. To enter a new procedure, type a P in the Type entry (the rest of the word is filled in for you) or press F6 for Choices and make your selection. Enter the name of the procedure, a description, and press F8. [Figure 9-12](#) shows a sample of this window.

Working:NEWAPP	
COPY MOVE SEARCH REPLACE	
Procedure Name: NEWDATA RQ5000	
Re-init Temps:	
;	
; CALCULATE AND DISPLAY THE NEW SALARY AND THE PERCENTAGE INCREASE	
;	
NEWSAL:FIELD P 4 2	;TEMP FIELD TYPE=PACKED, LEN=4, DEC PLACES = 2
PCT: FIELD F 2 2	;TEMP FIELD TYPE=FIXED, LEN=2, DEC PLACES = 2
;	
LET T.NEWSAL = OLDSAL + AMOUNT	;NEW SALARY
LET T.PCT = (AMOUNT/OLDSAL) * 100	;PERCENT INCREASE
;	
TRANSFER TO OUTPUT	;OUTPUT THE LINE
Required Name	

Figure 9-12 Procedure Definition Window

The Procedure Definition window consists of a header section that specifies procedure processing requirements and a detail section below the header section. The detail section is used to enter ASL statements. The ASL statements are described in the ASL Reference Guide.

Unless a procedure consists of only FIELD statements, it must be referenced elsewhere in the application in *exactly one* of the following ways:

- As an Input Control Procedure in a Screen Flow Conditions window.
- In an OUTPUT statement within a Conditions window.
- In a CALL statement in some other procedure.

Request Definition Window

The Request Definition window is accessed from the Application Definition window. To open the Request Definition window, either select an existing request or enter a new one. To enter a new request, type an R in the Type entry (the rest of the word is filled in for you) or press F6 for Choices and make your selection. Enter the name of the request, a description, and press F8. [Figure 9-13](#) shows a sample of this window.

Working:NEWAPP										
COPY MOVE SEARCH REPLACE										
Request Name: REQUESTA								PR5000		
Temp. Fields...: No					Re-init Temps...:					
Comments.....: No					Back Branch Ctl.:					
Seq	L	C	Q	Operand	O	Q	Operand	Q	Operand	Partial
/	U	O	L	...A....	P	L	...B....	L	...C....	S N O
Lbl	L	N	F	Name	R	F	Name/Constant	F	Name	T O P
				SALARY	LT	D	30000			
					NS		END			
				SALARY	*	D	1.10		T	NEWSAL
Required Name										

Figure 9-13 Request Definition Window

The Request Definition window can be used in place of the Procedure Definition window which uses the free-form procedural language, ASL. On the Request Definition window, you can enter requests consisting of fixed format PR and TF statements.

Unless a request consists of only temporary field definitions, it must be referenced elsewhere in the application in *exactly one* of the following ways:

- As an Input Control Procedure in a Screen Flow Conditions window.
- On an OUTPUT statement within a Conditions window.
- On a CALL statement in some procedure.

To define temporary fields from this window, put the cursor on Temp Fields button field and press F8. This opens a window where you can define temporary fields. Press F3 to close the window.

Screen Layout, Contents, and Actions

This chapter describes creating screen objects. A screen object identifies the layout, contents, and actions associated with a specific screen. Screen conditions are not part of the screen object and are discussed in [Chapter 9, *Creating VISION:Transact Applications*](#).

Screen objects can be accessed in two ways. The windows that define a screen and associated information are the same, regardless of the method you use.

The first way to access a screen is from the lower portion of the Application Definition window (as with procedures and requests). Screens created in this manner become objects but are not connected into the application. They must be connected into the application using the screen flow.

The second way to access a screen object is from the screen flow chart. Screens created from the screen flow chart are automatically connected into the application. To move to the screen flow chart from the Application Definition window, place the cursor on Screen Flow and press F8. A screen flow chart is shown in [Figure 10-1](#).

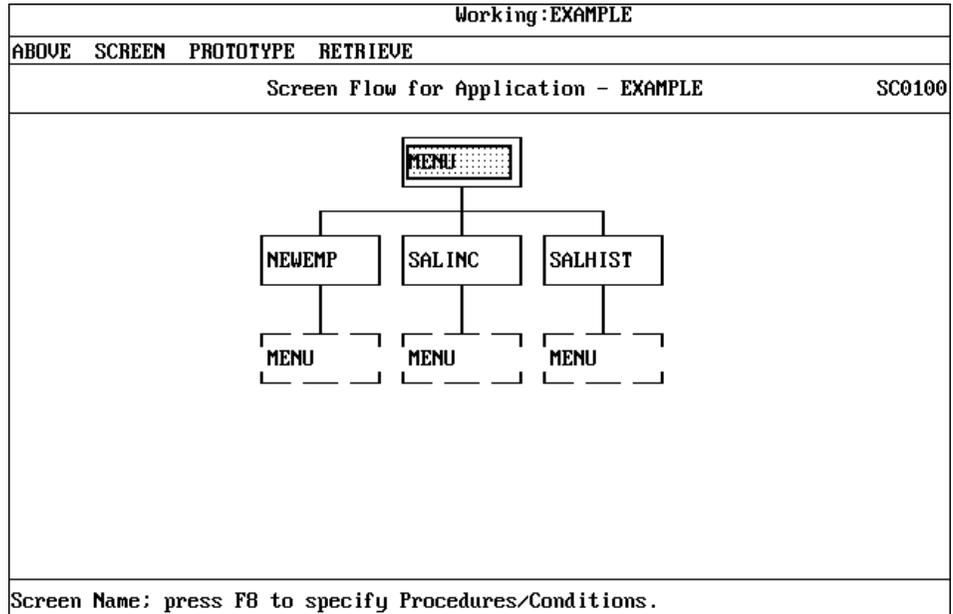


Figure 10-1 Screen Flow Chart

From the screen flow diagram place the cursor on the box that represents the screen you want to design and press F4 to move to the menu bar area. Select SCREEN and the Screen Definition window is displayed. Move to Layout and press F8.

Screen Layout

On the Screen Layout window, create the screen by building an image of the screen the way that it is to appear when the application is executed. Text is entered by typing the text where you want it to appear. A sample is shown in [Figure 10-2](#).

Working:EXAMPLE			
CENTER	EDIT	ATTRIBUTES	X=CURSOR
Line	1	Col 1	SCREEN LAYOUT - SALINC SD0200
<p>ENTER THE SALARY INCREASE FOR EMPLOYEE: _____</p> <p style="text-align: right;">FIRST NAME: _____ LAST NAME: _____</p> <p style="text-align: center;">DATE: _____ AMOUNT: _____</p> <p>TYPE OF INCREASE: _____</p> <p style="text-align: right;">PRESS PF10 WHEN DATA HAS BEEN ENTERED</p>			
Screen SALINC: press F8 to define screen field.			

Figure 10-2 Beginning the Screen Layout

On the Screen Layout window, the Tab key moves from field to field (left to right, top to bottom) and Shift Tab moves in the reverse direction. The Home key positions the cursor at the top left of the screen while End positions the cursor at the bottom left.

Fields are entered into the layout by positioning the cursor where the field belongs and pressing the OPEN key (F8). (Keep in mind that each data field must be preceded by and followed by a blank.) When the OPEN key is pressed, the Contents window is displayed. The Contents window is discussed in a later section.

Screen Layout Menu Bar Items

While designing the screen layout, the following menu bar items are available:

CENTER	Is a toggle that specifies whether the screen lines are to be centered.
EDIT	Activates another menu bar which displays the items EXCHANGE, COPY, MOVE, SEARCH, and REPLACE.
ATTRIBUTES	Lets you specify or change the display attributes.
CURSOR	Specifies the cursor position when the application is executed.

Using CENTER

The CENTER menu bar item works as a toggle on the Screen Layout window. It determines whether screen lines are to be centered. The toggle works in the following way:

- Toggle ON – Screen lines are centered.
- Toggle OFF – Screen lines are not centered.

You can tell whether it is on or off by its brightness. The CENTER menu bar item is highlighted when centering is on; it is dim when centering is off.

When centering is off (the default), you can position text where you want it to appear. You turn on centering by selecting CENTER: the menu bar disappears and you can return to entering the screen layout.

To center an existing screen line:

1. Turn on centering if it is not already on.
2. Type over at least one position in each line of the screen to be centered.
3. Press Enter. The screen lines are automatically centered.

Using EDIT

Selecting EDIT on the Screen Layout window causes the EXCHANGE, COPY, MOVE, SEARCH, and REPLACE menu bar to be displayed. EXCHANGE, COPY, and MOVE are block operations. You establish a block by marking the upper left corner followed by the lower right corner in response to the prompts when the operation is selected. Screen fields cannot be split or overlapped.

EXCHANGE allows you to swap the contents of two equal size areas on the screen.

MOVE allows you to move blocks of literal text, fields, and attributes. What is left from the move is blank text with the default attributes.

COPY allows you to copy text, fields, and literals; it leaves the original source data undisturbed. COPY is the only way you can create multiple occurrences of a screen field.

The SEARCH menu bar item can be used to search for strings of text. Follow the directions appearing at the bottom of the window to search for the text.

The REPLACE menu bar item can be used to search for a string and replace it with the replacement string that you specify. Follow the directions appearing at the bottom of the window to search for and replace the text when found.

Using ATTRIBUTES

The ATTRIBUTES menu item, in conjunction with the PAINT menu item, allows you to change the display attributes at the point at which the cursor is currently positioned. When you select this menu item, the Attributes window, shown in [Figure 10-3](#), is opened.

Working:EXAMPLE			
PAINT			
Line	Col	Text	Code
1	1	SCREEN LAYOUT - SALINC	SD0200
<p>ENTER THE SALARY INCREASE FOR EM</p> <div style="border: 1px solid black; padding: 5px; display: inline-block;"> <p style="text-align: center;">Change Display Attributes</p> <p>Color. RED Intensity.: Hilite: Pgm Symbol:</p> </div> <p>DATE: _____ AMOUNT: _____ TYPE OF INCREASE: _____</p> <p style="text-align: center;">PRESS PF10 WHEN DATA HAS BEEN ENTERED</p>			
CHOICES: <blank>, NEUTRAL, DEFAULT, BLUE, RED, PINK, GREEN, TURQUOISE, YELLOW			

Figure 10-3 Attributes Window

The most recently assigned attributes are displayed in the window. To change an attribute, make an entry or select an entry after pressing CHOICES (F6). Note that it is illegal for an attribute block with Intensity = Dark to encompass text. After filling in the window with the desired attributes, press F4 to move to the PAINT menu item. The PAINT menu item, discussed below, is used to actually change the screen layout attributes.

You can SWITCH (Alt+F8) to the Layout window to view the current attributes for various sections of the screen layout. After you press SWITCH, as you move the cursor around your screen layout, the Attributes window changes to display the current attributes for the cursor position.

Wherever possible, the PC monitor displays the screen layout as it would appear on a 3270 terminal. For example, if a portion of the screen is supposed to be red, it will be red. Those attributes which cannot be reasonably simulated by the PC monitor are displayed in reverse video. The DARK intensity attribute is displayed in reverse video during screen painting, but is displayed as DARK during prototyping.

Using PAINT

While the Attributes window is open, the PAINT menu item is displayed. The PAINT menu item is a block operation used to change the screen layout attributes to those in the Attributes window. As the block is marked, screen attributes are changed to the values in the Attributes window. Note that all occurrences of a field must have the identical intensity attribute; otherwise, an error message is issued during validation.

Using CURSOR

When CURSOR is selected, your cursor moves to the last cursor position specified (if there was one). Move the cursor (using Tab and BackTab) to where you want the cursor positioned when the screen is displayed during application execution. The last cursor position is lost when a new cursor position is specified. The position to which the VISION:Transact cursor is assigned is shown in inverse attributes.

Contents Window

The Contents window contains information on the particular screen field. To access the Contents window, place the cursor within the selected field and press F8. Because structured screens can have repeated segments, there may be more than one representation of the field on the screen. Any representation (repeated field) can be used to open the window. Changes made to the window apply to all representations; all repeats of the same field have the same window. A Contents window is shown in [Figure 10-4](#).

Working:EXAMPLE			
COPY	MOVE	SEARCH	REPLACE
Line	5	Col 17	SD0200
SCREEN LAYOUT - MENU			
CHOOSE A SCREEN BY USING THE FOLLOWING FUNCTION KEYS			
Segment.....: 1	Name:EMPLNO	Type..:INPUT	Length.:
INPUT-->To...: EMPLNO		Action: MATCH SEGMENT	
Hilite Err...:	Num. Lock:	Display Modifications:	Protect:
OUTPUT-->From:			
EITHER-->Num. Edit:		Edit Mask:	
— Additional Validation/Actions			
Name of screen field			

Figure 10-4 Contents Window

The top half of the Contents window provides information about the display/entry of data in the field.

Name is required.

Length is usually optional. When entered, length overrides the calculated number of screen positions required by the screen field. Length is only required for virtual fields without a destination. The full length of the field must fit on the screen line; fields are not permitted to fold. If the length of the screen field is not specified, the order of precedence in which the length is calculated is:

- The length of the input field in the file definition.
- The length of the output field in the file definition.
- One.

When the Contents window is closed, the information is accepted and the length of the field on the screen is calculated and underscores (the default) are provided to represent field values.

INPUT To and OUTPUT From identify the destination and source fields, respectively. Multiple destination fields are accommodated by the DO INPUT statement in the Additional Validation/Actions portion of the window. This is discussed below.

Action specifies the action to be performed with the data in the screen input field. To view the available entries, press CHOICES (F6) and make your selection.

Hilite Error indicates whether the field is to be highlighted when the screen is re-displayed after an error is detected.

Num Lock sets the numeric field attribute, but no automatic numeric validation is performed.

Display Modifications specifies whether the user sees the original value in the screen field or if the user sees the value of the screen field after updates have occurred.

Protect is used to inhibit keyboard input into a screen field. The terminal operator cannot enter data into a protected field.

Numeric Edit allows you to specify whether grouping characters are used when displaying a numeric field.

Edit Mask allows you to specify VISION:Transact edit patterns.

Additional Validation/Actions

The bottom half of the Contents window contains the Additional Validation/Actions section. In this section, you enter the validation rules and the automatic actions that are to be taken as a result of data entry. This window is scrollable to allow for lengthy validation/actions.

In this section, you can enter a DO INPUT block for each destination field. A DO INPUT block begins with a DO statement and ends with an END statement. It can contain the following statement types:

DO statement – begins the block and identifies the destination field to which the action(s) apply. The destination field name is specified in the form of qualifier.fieldname or a PF function. The format of the DO statement is:

```
DO INPUT FOR fieldname
```

If the destination field is not specified in the DO INPUT statement, it is assumed to be the destination field entered in the top part of the window, such as the INPUT To: field. In this case, the format of the DO statement is:

```
DO INPUT
```

VERIFY statement – can follow the DO INPUT statement to specify field level validation. This statement is equivalent to the VISION:Transact fixed format Sn validation operators. The VERIFY statement must precede any CASE statements and must follow the DO statement. The format of the VERIFY statement is:

```
VERIFY FIELD verify-expression MESSAGE 'message-text'
```

Verify-expression is a logical expression specifying the screen field name (with a PF function if required) on the left side of each term.

Message-text is the text of the message to be displayed if verify-expression is false.

For example:

```
VERIFY S.INPUT1 NE '' AND PF(S.INPUT1,3,1) EQ 'X',  
MESSAGE 'INPUT MISSING OR INVALID FORMAT'
```

Action statement – identifies the action(s) to apply to the destination field. The following terms are valid (the word SEGMENT is optional):

DELETE SEGMENT	ADD
INSERT SEGMENT	SUBTRACT
MATCH SEGMENT	REPLACE
UPDATE SEGMENT	REPLACE IF NONBLANK
UPDATE OR INSERT	CLEAR
REPEAT	NOACTION
IGNORE	

CASE statement – begins the CASE block. It is equivalent to the VISION:Transact IF operator. The format of the CASE statement is:

```
CASE control-field operator value
```

Control-field is any screen field (qualifier S) or system field (qualifier F).

Operator is EQ or NE.

Value is a constant or system field (qualifier F).

ON statement – specifies action to be taken in the event of certain error conditions. The ON statement must precede any CASE or ACTION statements. There are two forms of the ON statement. The first form is used to override standard VISION:Transact error messages and is equivalent to the VISION:Transact EC/ET operators:

```
ON ERROR error-id MESSAGE 'message-text'
```

Error-id can be one of the following error conditions:

```
INVALID  
NOT_FOUND  
DUPLICATE  
NO_ROOM  
RECORD_OVERFLOW  
CONVERSION
```

Message-text is the text of the message to be displayed upon the error condition.

The second form of the ON statement is used only with virtual fields and indicates the screen field(s) to be highlighted in the event of an error:

```
ON ERROR HILITE screen field(s)
```

ELSE statement – precedes the default action when specifying dynamic actions. The ELSE statement has no keywords or operands.

END statement – terminates a DO block. The END statement has no keywords or operands.

Comment statements – can be entered anywhere following a semicolon (;).

Destination Fields – An input screen field may have zero, one, or more destinations. For the sake of discussion, call one of these the *primary destination* and any others *alternate destinations*. The primary destination is named by the INPUT To entry in the upper part of the Contents window. Alternate destinations are named using the DO statement. You can also have a DO statement for the primary destination. In this case, the destination name can be omitted from the DO statement. Also, if you have an input field with *no* destination, the field name must be left blank in the DO statement.

Specifying Actions – If only a single action is required for a destination field, it can be specified using the Action entry in the upper part of the Contents window (primary destination) or in a DO block (alternate destination). (VISION:Workbench also allows you to specify an action for the primary destination within a DO block. In this case, the Action entry in the upper part of the window must be left blank.)

Dynamic actions for a destination field are preceded by a CASE statement. The first CASE condition which is true determines the action to be performed at execution time. A default action is always required. This is specified following the ELSE statement. (For the primary destination, you can optionally specify the default action using the Action entry in the upper part of the window instead of using an Action statement following an ELSE statement.)

The following example illustrates the general structure of the Output Conditions section together with each of the available statement types:

```
DO INPUT
VERIFY S.CUSTNO NE ' ' MESSAGE 'CUSTOMER NUMBER REQUIRED'
ON NOT_FOUND MESSAGE 'CUSTOMER NOT ON DATABASE'
UPDATE SEGMENT
END
;
DO INPUT FOR 1.CUSTNO
CASE F.ATTNID EQ 'PF001'
    MATCH SEGMENT
CASE F.ATTNID EQ 'ENTER'
    UPDATE SEGMENT
ELSE
    IGNORE
END
```

This example has two destination fields for the same screen field. (The name of the first destination appears in the upper part of the Contents window and it is not shown here.) In the example, the screen field is validated and, if it is blank, the message “Customer Number Required” is output to the user's terminal. If a customer number is entered but no database segment exists for that customer, the message “Customer Not on Database” (instead of a standard system error message) is displayed. UPDATE SEGMENT identifies the action to be performed. (Note that this action could just as well have been specified in the Action entry on the upper part of the Contents window.)

The second destination field, 1.CUSTNO, illustrates the use of dynamic actions. The first CASE condition which is true determines the action to be performed.

Note that all VERIFY and ON statements must precede any CASE or action statements.

Walk-Through for VISION:Transact Applications

This chapter lets you get a feel for VISION:Workbench by actually using it. A VISION:Transact application has been predefined in the library. The changes you are asked to make to the application may not seem realistic; they are only intended for you to use as much of VISION:Workbench as possible.

For the best results, enter the steps of this walk-through on your PC. However, this chapter contains copies of every window and PC interaction and can be read without the use of a PC.

As you work your way through the application, you select menu items, enter information, use the function keys, and use the menu bar. Each step of the way, the action to be performed in the particular window is described and an illustration of the window is included both before and after the action is taken. If the action results in another window being opened, that window is shown instead. Your window should always resemble the illustrations in this chapter.

Throughout most of this walk-through, the window you are looking at on your PC is shown on the top half of the page; the action you must take is on the bottom half of the page.

The walk-through concludes by having you export the application. Once exported, the application can be uploaded to the host where it can be subsequently processed.

Application Overview

This application is a simple personnel application. There are two files involved in this application, a personnel file (PERSFILE) and a department file (DEPTFILE).

The personnel file is a two-level file, as shown in [Figure 11-1](#).

EMPLNO	FIRSTNM	LASTNM	DEPTNO	SALARY	TITLE
Employee Number	First Name	Last Name	Department Number	Salary	Employee Title
		DATE	AMOUNT	OLDSAL	TYPEINC
		Date of Increase	Amount of Increase	Previous Salary	Type of Increase

Figure 11-1 The Personnel File

The department file is a single-level structure, as shown in [Figure 11-2](#). This file is not used in the current application, but, as you will see in the walk-through, you have been asked to add data from it.

DEPTNUM	DEPTDESC	LOCATION	BUDGET
Department Number	Description	Location	Annual Budget

Figure 11-2 The Department File

The application consists of the four screens illustrated in [Figure 11-3](#).

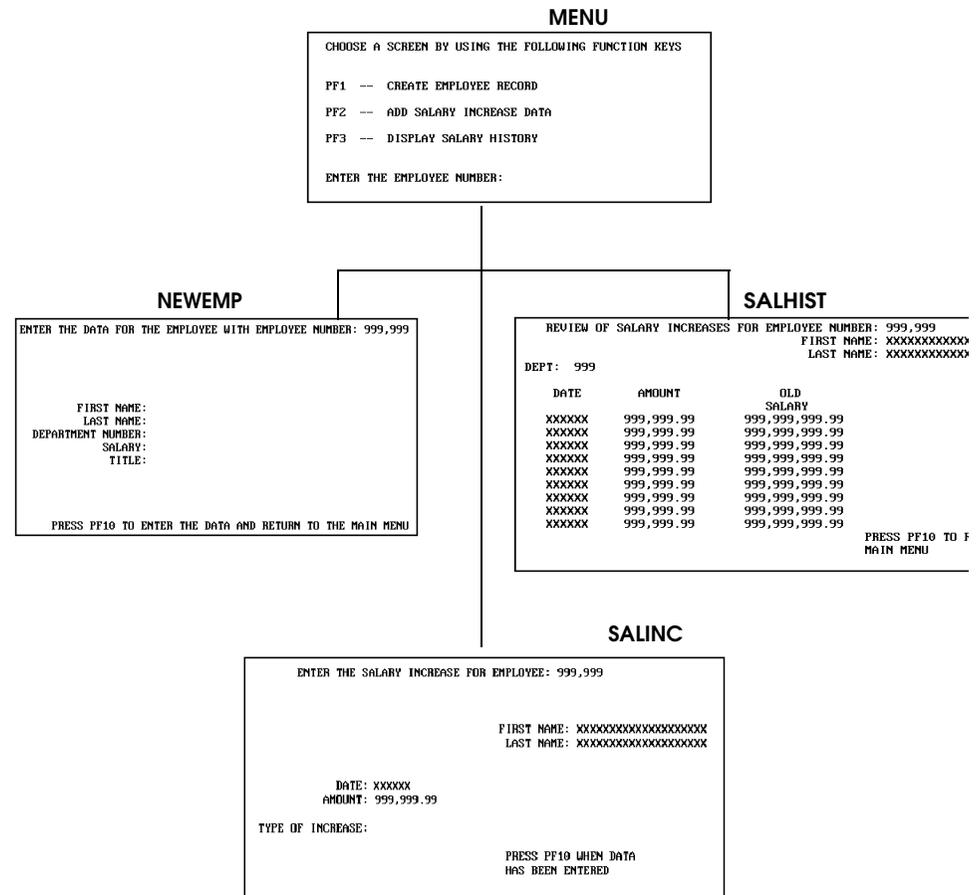


Figure 11-3 Application Screens

- MENU** Is a simple menu for the user to choose an option.
- NEWEMP** Allows the creation of a new record for a new employee, such as the root segment for PERSFILE.
- SALINC** Provides for recording a salary increase, such as the low-level segment of PERSFILE.
- SALHIST** Provides a review of all of the increases received by an individual.

The users want the SALHIST screen to be changed. The current screen is shown in [Figure 11-4](#).

Working:EXAMPLE				
CENTER EDIT ATTRIBUTES X=CURSOR				
Line	1	Col	1	SD0200
SCREEN LAYOUT - SALHIST				
REVIEW OF SALARY INCREASES FOR EMPLOYEE NUMBER: _____				
FIRST NAME: _____				
LAST NAME: _____				
DEPT: _____				
DATE	AMOUNT	OLD SALARY		
_____	_____	_____		
_____	_____	_____		
_____	_____	_____		
_____	_____	_____		
_____	_____	_____		
_____	_____	_____		
_____	_____	_____		
_____	_____	_____		
PRESS PF10 TO RETURN TO THE MAIN MENU				
Screen SALHIST: press F8 to define screen field.				

Figure 11-4 Current SALHIST Screen

The users have requested that you change this screen to be as shown in [Figure 11-5](#).

Working:EXAMPLE				
CENTER EDIT ATTRIBUTES X=CURSOR				
Line	1	Col	1	SD0200
SCREEN LAYOUT - SALHIST				
REVIEW OF SALARY INCREASES FOR EMPLOYEE NUMBER: _____				
FIRST NAME: _____				
LAST NAME: _____				
DEPT: _____				
DATE	AMOUNT	OLD SALARY	NEW SALARY	PERCENT INCREASE
_____	_____	_____	_____	_____
_____	_____	_____	_____	_____
_____	_____	_____	_____	_____
_____	_____	_____	_____	_____
_____	_____	_____	_____	_____
_____	_____	_____	_____	_____
_____	_____	_____	_____	_____
_____	_____	_____	_____	_____
PRESS PF10 TO RETURN TO THE MAIN MENU				
Screen SALHIST: press F8 to define screen field.				

Figure 11-5 Updated SALHIST Screen

Further, because they have just installed color terminals, the users would like the Percent Increase column to be displayed in red.

Also, the data processing organization has just adopted new standards which require that PF3 not be used in an application. You have been requested to use PF4 instead.

In summary, here are the changes you must make:

1. Change the use of PF3 to PF4. This actually requires three changes:
 - The logic of the application must be changed so that the PF4 key is used instead of PF3.
 - The MENU screen must be changed so that the text describes PF4 instead of PF3.
 - The procedure (CHECKPF) that checks that only PF1, PF2, and PF3 were pressed on the menu display must be changed to check for PF1, PF2, and PF4.
2. Change the SALHIST screen to Department Name instead of Department Number. This will require integrating the DEPTFILE into the application.
3. Compute the new salary and the percentage of increase.
4. Display New Salary and Percentage of Increase columns on the SALHIST screen.
5. Display the Percentage Increase column in red.

Important Keys

There are several keys that you should be familiar with before beginning the application. The following keys and function key descriptions are used for navigation and operation of VISION:Workbench:

Function Key	Description
Enter (Return)	Allows you to select an item from a Choices list or to complete the entry of an item.
F1	Displays the function key bar which provides you with the meaning of the available function keys for the current window.
F3	Indicates that you are finished with a window. It closes the window and takes you back to the higher level window. It also terminates processing when pressed from the main menu.
F4	Moves the cursor to the menu bar which displays the menu items available for the current window.
F8	Opens a window where you can provide or display more information on the current item.

Function Key	Description
Tab	On the Screen Layout Definition window, moves the cursor from field to field (left to right, top to bottom). Shift-Tab moves the cursor in the reverse direction.
Home	On the Screen Layout Definition window, positions the cursor at the top left of the screen.
End	Positions the cursor at the bottom left.

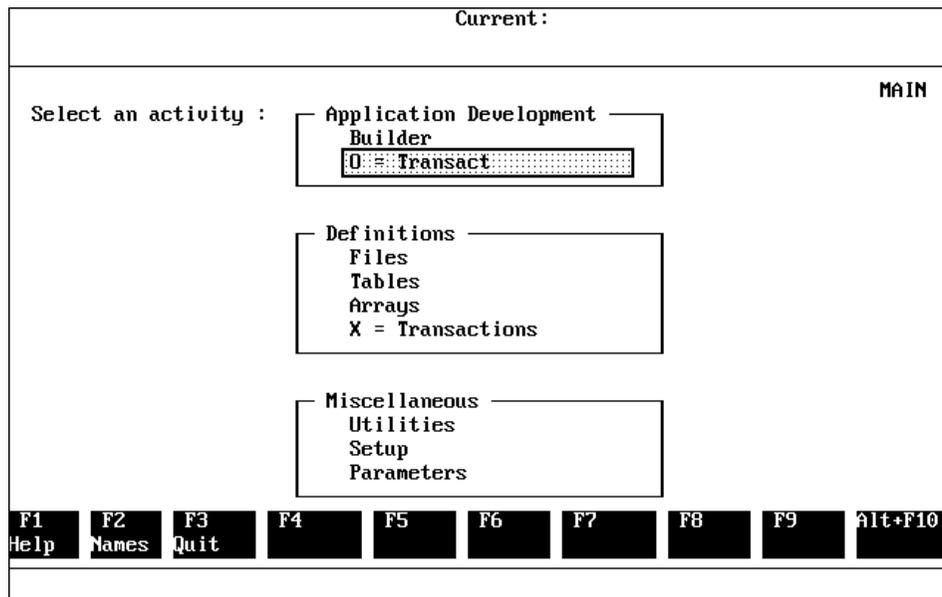
Beginning the Application

Turn your PC on and go to the directory containing the VISION:Workbench for DOS.

Type mpcx and press Enter.

The Primary Library Path on the Setup window must be specified as \MPC55\SAMPAPPS to access the sample application. See Specifying Setup Information in [Chapter 2, Getting Started – Installation and Setup](#) for information on the Setup Window.

The Banner window is displayed first. Press any key to move to the VISION:Workbench for DOS Copyright window (not shown). Press any key again to move to the Main Menu.



This is the Main Menu. You want to start by updating a VISION:Transact application. Move your cursor to O = Transact and press F8 or simply press O for Transact.

Follow the instructions on the bottom line of the screen (watch the bottom line; it always provides a helpful prompt).

VISION:Workbench for DOS		Current :
USERINFO	User file index information	FI0100
Type	Name	Location
TRANSACT	EXAMPLE	C:\MPCXX\SAMPAPPS
Name of item; press F8 to select this item.		

Generally, you would have a list of Transact applications.

This example shows only one.

Position the cursor over the word EXAMPLE and press F8.

VISION:Workbench for DOS		Working:EXAMPLE	
RETRIEVE X=EXPORT PROTOTYPE			
Application Name: EXAMPLE			RC5000
Transact Application Sections			
Files	: Yes	Screen Flow:	Yes Environment: No Comments: No
Screens	: 4	Procs	: 3 Requests : 0
Type	Name	Share	Description
SCREEN	MENU	NO	INITIAL MENU SCREEN
SCREEN	NEWEMP	NO	ENTER EMPLOYEE DATA
SCREEN	SALINC	NO	PROVIDE SALARY INCREASE DATA
SCREEN	SALHIST	NO	DISPLAY THE SALARY HISTORY
PROC	CHECKPF	NO	CHECK THAT ONLY PF1, PF2 OR PF3 PRESSED
PROC	CHEKPF10	NO	CHECK THAT ONLY PF10 PRESSED
PROC	SALCALC	NO	CALCULATE SALARIES FROM INCREASE DATA
Required Name			

Here is the application with all its components displayed.

The Share column tells you whether another application can use your screen or procedure.

Because several people may want to practice with this example, please do not destroy the original example.

Create a copy of the application by simply changing the name from EXAMPLE to NEWAPP.

VISION:Workbench for DOS		Working:EXAMPLE	
RETRIEVE X=EXPORT PROTOTYPE			
Application Name: <input type="text" value="NEWAPP"/>			RC5000
Transact Application Sections			
Files	: Yes	Screen Flow	: Yes
Environment	: No	Comments	: No
Screens	: 4	Procs	: 3
Requests	: 0		
Type	Name	Share	Description
SCREEN	MENU	NO	INITIAL MENU SCREEN
SCREEN	NEWEMP	NO	ENTER EMPLOYEE DATA
SCREEN	SALINC	NO	PROVIDE SALARY INCREASE DATA
SCREEN	SALHIST	NO	DISPLAY THE SALARY HISTORY
PROC	CHECKPF	NO	CHECK THAT ONLY PF1, PF2 OR PF3 PRESSED
PROC	CHEKPF10	NO	CHECK THAT ONLY PF10 PRESSED
PROC	SALCALC	NO	CALCULATE SALARIES FROM INCREASE DATA
Required Name			

All future actions will modify NEWAPP and leave the original EXAMPLE undisturbed.

Before starting, take a moment to get a feel for the application by prototyping it. Only a valid application can be prototyped, so first you must validate the application.

VISION:Workbench for DOS		Validate:NEWAPP	
RETRIEVE X=EXPORT PROTOTYPE			
Application Name:NEWAPP			RC5000
Transact Application Sections			
Files	: Yes	Screen Flow:	Yes Environment: No Comments: No
Screens	: 4	Procs	: 3 Requests : 0
Type	Name	Share	Description
SCREEN	MENU	NO	INITIAL MENU SCREEN
SCREEN	NEWEMP	NO	ENTER EMPLOYEE DATA
SCREEN	SALINC	NO	PROVIDE SALARY INCREASE DATA
SCREEN	SALHIST	NO	DISPLAY THE SALARY HISTORY
PROC	CHECKPF	NO	CHECK THAT ONLY PF1, PF2 OR PF3 PRESSED
PROC	CHEKPF10	NO	CHECK THAT ONLY PF10 PRESSED
PROC	SALCALC	NO	CALCULATE SALARIES FROM INCREASE DATA
Validation complete - no errors - Press any key.			

Press F1 to display the function key bar and F7 to validate the application. (You do not need to press F1 if you remember that F7 activates validation.) Press any key when the validation is complete.

Press F4 to move the cursor to the menu bar. Move the cursor to PROTOTYPE and press Enter.

VISION:Workbench for DOS		Validate:NEWAPP	
RETRIEVE X=EXPORT PROTOTYPE			
Application Name:NEWAPP			RC5000
Transact Application Sections			
Files	: Yes	Screen Flow:	Yes Environment: No Comments: No
Screens	: 4	Procs	: 3 Requests : 0
Type	Name	Share	Description
SCREEN	MENU	NO	INITIAL MENU SCREEN
SCREEN	NEWEMP	NO	ENTER EMPLOYEE DATA
SCREEN	SALINC	NO	PROVIDE SALARY INCREASE DATA
SCREEN	SALHIST	NO	DISPLAY THE SALARY HISTORY
PROC	CHECKPF	NO	CHECK THAT ONLY PF1, PF2 OR PF3 PRESSED
PROC	CHEKPF10	NO	CHECK THAT ONLY PF10 PRESSED
PROC	SALCALC	NO	CALCULATE SALARIES FROM INCREASE DATA
PC simulation of screen interaction			

You can walk through the application by pressing F1, F2, F3, and F10 for PF1, PF2, PF3, and PF10. Pressing other F keys will terminate the application prototyping.

VISION:Workbench for DOS		Working: NEWAPP	
RETRIEVE X=EXPORT PROTOTYPE			
CHOOSE A SCREEN BY USING THE FOLLOWING FUNCTION KEYS			
PF1 -- CREATE EMPLOYEE RECORD			
PF2 -- ADD SALARY INCREASE DATA			
PF3 -- DISPLAY SALARY HISTORY			
ENTER THE EMPLOYEE NUMBER:			
Press <ESC> to choose follow-on screen			

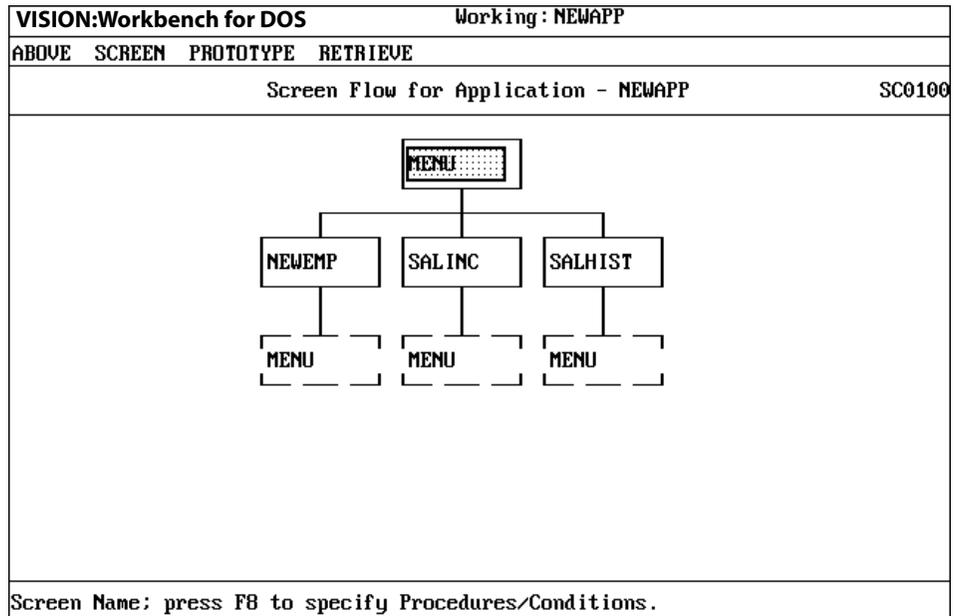
An alternative way to display a follow-on screen is given by the prompt at the bottom of the screen. Press Esc and select the follow-on screen.

When you have a feel for the application, press Esc and F3 to return to the Application Definition window.

VISION:Workbench for DOS		Validate:NEWAPP																																	
RETRIEVE X=EXPORT PROTOTYPE																																			
Application Name:NEWAPP			RC5000																																
Transact Application Sections																																			
Files	: Yes	Screen Flow	: Yes																																
Environment	: No	Comments	: No																																
Screens	: 4	Procs	: 3																																
Requests	: 0																																		
<table border="1"> <thead> <tr> <th>Type</th> <th>Name</th> <th>Share</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>SCREEN</td> <td>MENU</td> <td>NO</td> <td>INITIAL MENU SCREEN</td> </tr> <tr> <td>SCREEN</td> <td>NEWEMP</td> <td>NO</td> <td>ENTER EMPLOYEE DATA</td> </tr> <tr> <td>SCREEN</td> <td>SALINC</td> <td>NO</td> <td>PROVIDE SALARY INCREASE DATA</td> </tr> <tr> <td>SCREEN</td> <td>SALHIST</td> <td>NO</td> <td>DISPLAY THE SALARY HISTORY</td> </tr> <tr> <td>PROC</td> <td>CHEKPF</td> <td>NO</td> <td>CHECK THAT ONLY PF1, PF2 OR PF3 PRESSED</td> </tr> <tr> <td>PROC</td> <td>CHEKPF10</td> <td>NO</td> <td>CHECK THAT ONLY PF10 PRESSED</td> </tr> <tr> <td>PROC</td> <td>SALCALC</td> <td>NO</td> <td>CALCULATE SALARIES FROM INCREASE DATA</td> </tr> </tbody> </table>				Type	Name	Share	Description	SCREEN	MENU	NO	INITIAL MENU SCREEN	SCREEN	NEWEMP	NO	ENTER EMPLOYEE DATA	SCREEN	SALINC	NO	PROVIDE SALARY INCREASE DATA	SCREEN	SALHIST	NO	DISPLAY THE SALARY HISTORY	PROC	CHEKPF	NO	CHECK THAT ONLY PF1, PF2 OR PF3 PRESSED	PROC	CHEKPF10	NO	CHECK THAT ONLY PF10 PRESSED	PROC	SALCALC	NO	CALCULATE SALARIES FROM INCREASE DATA
Type	Name	Share	Description																																
SCREEN	MENU	NO	INITIAL MENU SCREEN																																
SCREEN	NEWEMP	NO	ENTER EMPLOYEE DATA																																
SCREEN	SALINC	NO	PROVIDE SALARY INCREASE DATA																																
SCREEN	SALHIST	NO	DISPLAY THE SALARY HISTORY																																
PROC	CHEKPF	NO	CHECK THAT ONLY PF1, PF2 OR PF3 PRESSED																																
PROC	CHEKPF10	NO	CHECK THAT ONLY PF10 PRESSED																																
PROC	SALCALC	NO	CALCULATE SALARIES FROM INCREASE DATA																																
Press F8 to select this item																																			

The first task is to change the usage of the PF3 key. To do this, you need to locate the screen activated by PF3 from the Menu screen.

Position the cursor on Screen Flow and open the Screen Flow window by pressing F8.

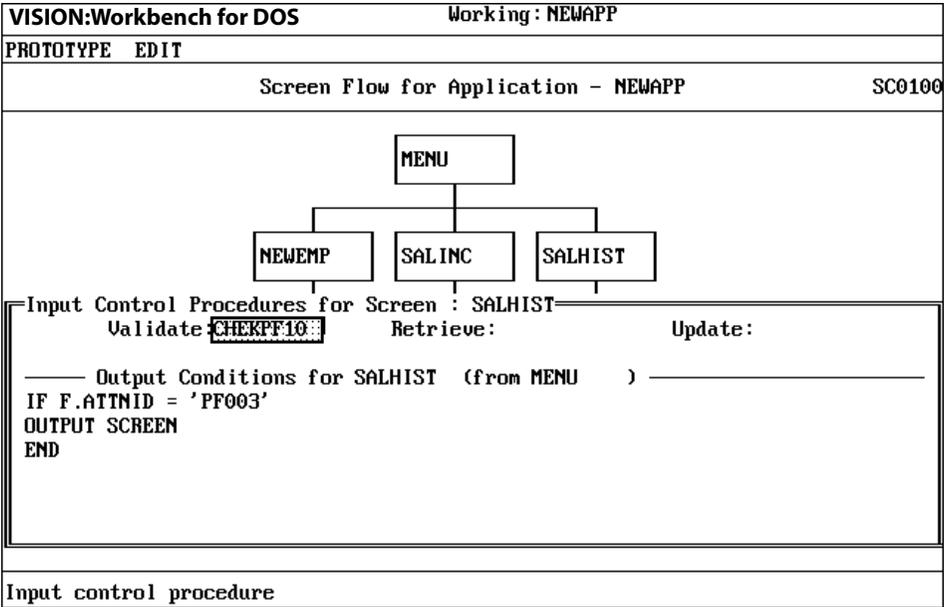


The screen flow shows what output screens can be displayed as follow-on screens from other screens.

To open the Conditions window for the SALHIST screen, position the cursor on SALHIST (down arrow and Tab) and press F8.

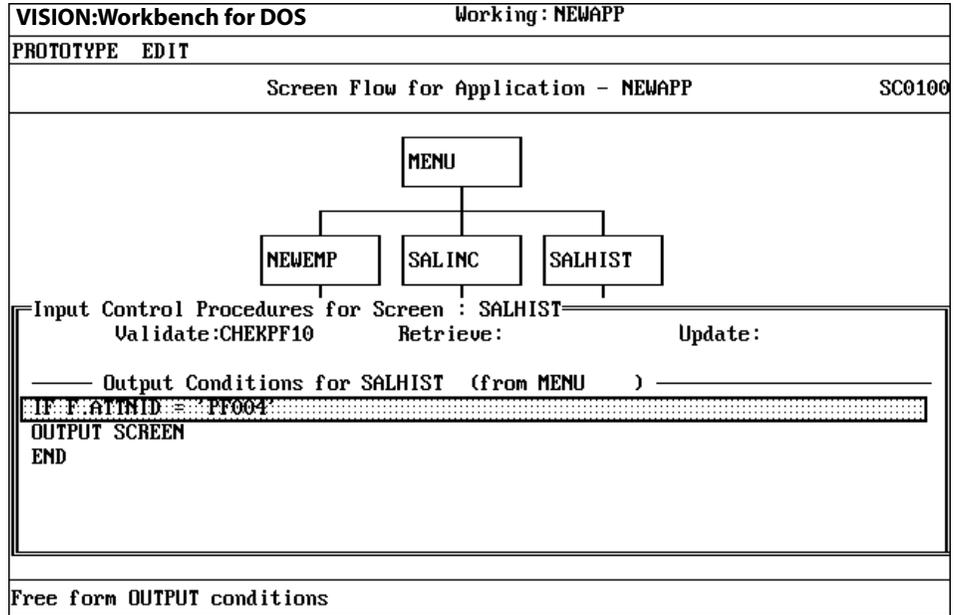
If you want to review all of the conditions for each of the screens, do the following:

- 1. Press F8 on MENU to open the Conditions window.
- 2. Press Alt+F8 to switch the cursor out of the Conditions window.
- 3. Use the arrow keys and Tab to move the cursor across the boxes. The Conditions window changes for each box. If you accidentally open a blank box, just arrow back to an existing box. End up on SALHIST.
- 4. Press Alt+F8 to switch the cursor into the Conditions window.



The Conditions window opens and shows the procedures activated during the interaction of the MENU screen and the SALHIST screen.

Press PF3 to display the bottom part of the Conditions window, which shows the condition under which SALHIST will be displayed from MENU.

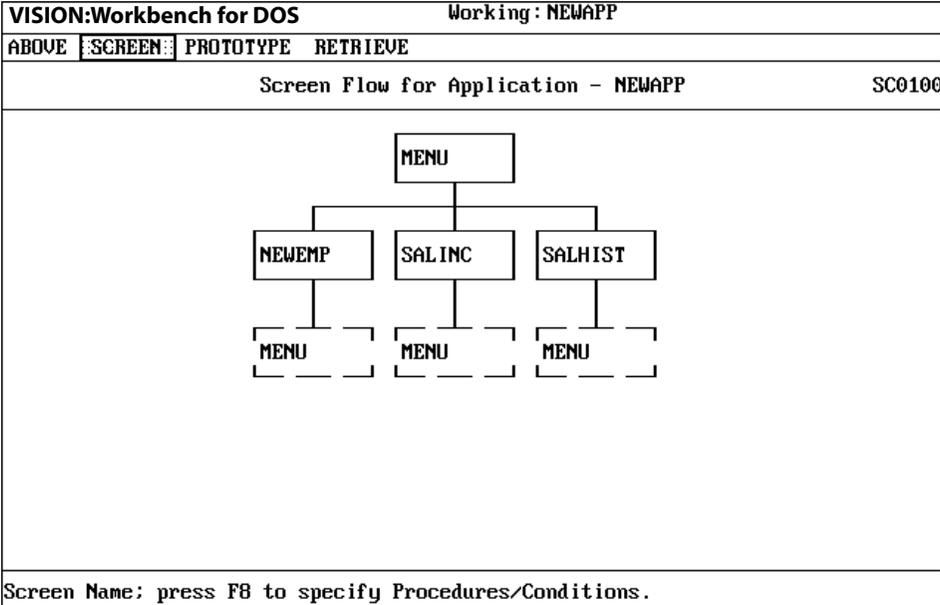


Change PF003 to PF004. You have now changed the application to use PF4 instead of PF3.

The next task is to change references on the MENU screen from PF3 to PF4.

Close the Conditions window by pressing F3.

Move the cursor to MENU and press F4 to move the cursor to the menu bar.



Move the cursor to SCREEN and press Enter to view the MENU screen structure.

Segment	Occurs	Display	Virtual Fields
1	1	1	0

The first part of the screen describes the screen structure.

Position the cursor over Layout and press F8 to open the Layout window (to show the screen layout).

VISION:Workbench for DOS				Working: NEWAPP	
CENTER EDIT ATTRIBUTES X=CURSOR					
Line	1	Col	1	SCREEN LAYOUT - MENU	SD0200
<p>CHOOSE A SCREEN BY USING THE FOLLOWING FUNCTION KEYS</p> <p>PF1 -- CREATE EMPLOYEE RECORD</p> <p>PF2 -- ADD SALARY INCREASE DATA</p> <p>PF3 -- DISPLAY SALARY HISTORY</p> <p>ENTER THE EMPLOYEE NUMBER: _____</p>					
Screen MENU; press F8 to define screen field.					

Position the cursor over the 3 and type 4.

VISION:Workbench for DOS				Working: NEWAPP	
CENTER EDIT ATTRIBUTES X=CURSOR					
Line	9	Col	15	SCREEN LAYOUT - MENU	SD0200
<p>CHOOSE A SCREEN BY USING THE FOLLOWING FUNCTION KEYS</p> <p>PF1 -- CREATE EMPLOYEE RECORD</p> <p>PF2 -- ADD SALARY INCREASE DATA</p> <p>PF4 -- DISPLAY SALARY HISTORY</p> <p>ENTER THE EMPLOYEE NUMBER: _____</p>					
Screen MENU; press F8 to define screen field.					

Press F3 three times:

- Once to close the Layout window.
- Once to close the Screen Structure window.
- Once to close the Screen Flow window.

VISION:Workbench for DOS		Working: NEWAPP	
RETRIEVE X=EXPORT PROTOTYPE			
Application Name: NEWAPP			RC5000
Transact Application Sections			
Files	: Yes	Screen Flow: Yes	Environment: No
Comments	: No	Requests	: 0
Screens	: 4	Procs	: 3
Type	Name	Share	Description
SCREEN	MENU	NO	INITIAL MENU SCREEN
SCREEN	NEWEMP	NO	ENTER EMPLOYEE DATA
SCREEN	SALINC	NO	PROVIDE SALARY INCREASE DATA
SCREEN	SALHIST	NO	DISPLAY THE SALARY HISTORY
PROC	CHECKPF	NO	CHECK THAT ONLY PF1, PF2 OR PF3 PRESSED
PROC	CHEKPF10	NO	CHECK THAT ONLY PF10 PRESSED
PROC	SALCALC	NO	CALCULATE SALARIES FROM INCREASE DATA

Press F8 to select this item

Back at the Main Menu, position the cursor on the CHECKPF Proc line.

This is the procedure that checks for PF1, PF2, and PF3. You must change it to check for PF4 instead of PF3.

Press F8 to open the Procedures window.

```
VISION:Workbench for DOS Working:NEWAPP
COPY MOVE SEARCH REPLACE
Procedure Name: CHECKPF RQ5000
Re-init Temps:
;
; THIS PROC CHECKS THAT THE CORRECT PF KEYS WERE PRESSED
; AND ISSUES A MESSAGE IF THE WRONG KEYS ARE PRESSED.
;
IF P.ATTNID NE 'PF001', 'PF002', 'PF003' THEN
MSG 'PRESS ONLY PF1, PF2 OR PF3'
END
Required Name
```

Position the cursor over each 3 and type 4.

```

VISION:Workbench for DOS           Working:NEWAPP
COPY  MOVE  SEARCH  REPLACE

Procedure Name: CHECKPF           RQ5000
Re-init Temps:

;
; THIS PROC CHECKS THAT THE CORRECT PF KEYS WERE PRESSED
; AND ISSUES A MESSAGE IF THE WRONG KEYS ARE PRESSED.
;
IF F.ATTNID NE 'PF001', 'PF002', 'PF004' THEN
MSG 'PRESS ONLY PF1, PF2 OR PF4'
END
    
```

Enter a Procedure Statement

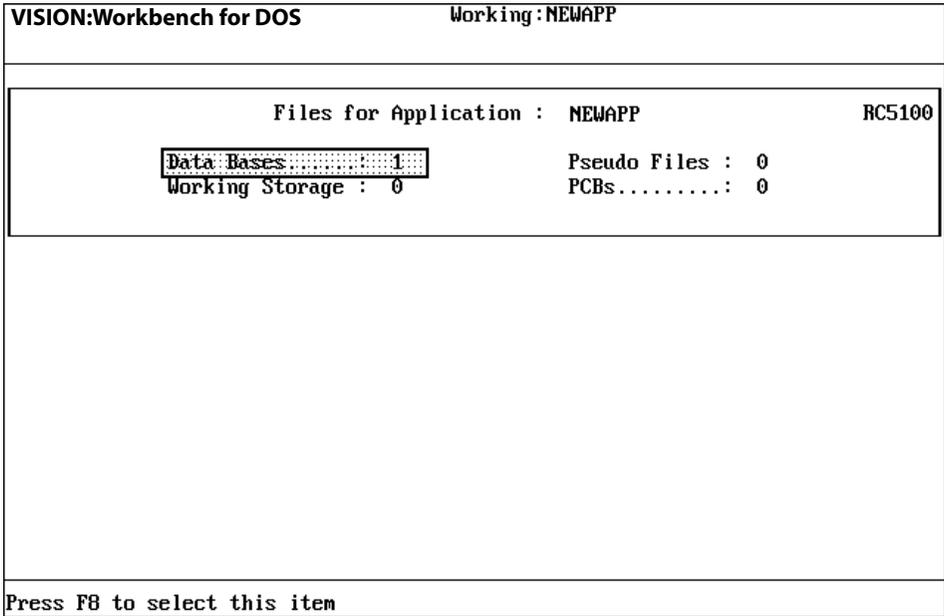
Press F3 to close the Procedures window.

VISION:Workbench for DOS		Working:NEWAPP	
RETRIEVE X=EXPORT PROTOTYPE			
Application Name: NEWAPP			RC5000
Transact Application Sections			
Files	Yes	Screen Flow: Yes	Environment: No
Comments:	No	Procs : 3	Requests : 0
Screens :	4		
Type	Name	Share	Description
SCREEN	MENU	NO	INITIAL MENU SCREEN
SCREEN	NEWEMP	NO	ENTER EMPLOYEE DATA
SCREEN	SALINC	NO	PROVIDE SALARY INCREASE DATA
SCREEN	SALHIST	NO	DISPLAY THE SALARY HISTORY
PROC	CHEKPF	NO	CHECK THAT ONLY PF1, PF2 OR PF4 PRESSED
PROC	CHEKPF10	NO	CHECK THAT ONLY PF10 PRESSED
PROC	SALCALC	NO	CALCULATE SALARIES FROM INCREASE DATA
Description of item; press F8 to select this item.			

Next, clean up the description of CHECKPF by changing PF3 to PF4.

The next task is to change the department number displayed on screen SALHIST to the department description. Because the department description exists in another file (DEPTFILE), you must add a file to our application.

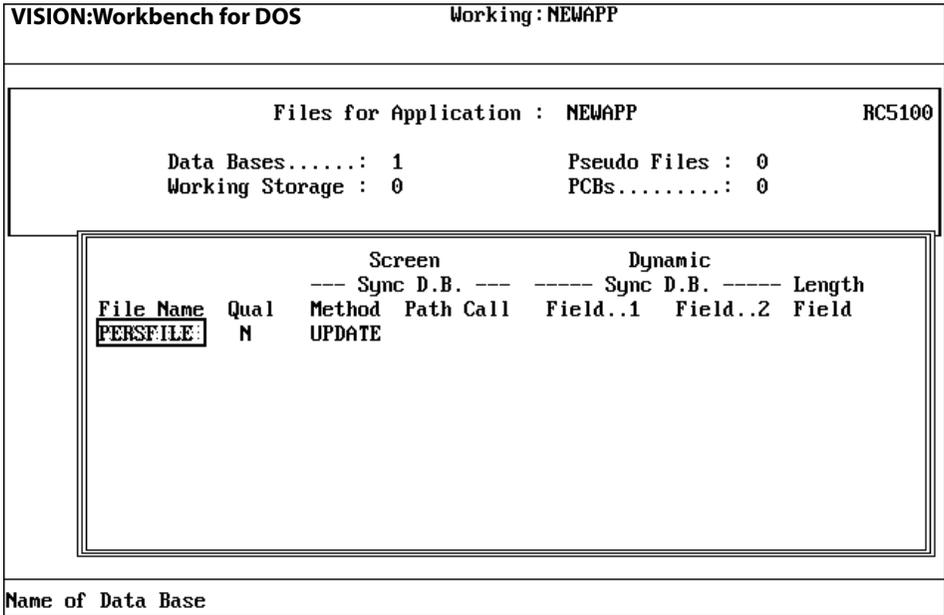
Position the cursor over Files and open the Files window by pressing F8.



The Files window shows you the types of files that are active in the application.

In this example, you only have one database.

To add a database, position the cursor over Data Bases and open the Data Base window by pressing F8.



The Data Base window is displayed.

Now, add DEPTFILE and give it the qualifier 1. Notice that the bottom line prompt tells you the permissible entries for the field where the cursor is positioned. In this case, you want to dynamically synchronize the DEPTFILE based on the DEPTNO field in the Personnel file. Enter DEPTNO as the synchronizing field.

```

VISION:Workbench for DOS           Working: NEWAPP
-----
Files for Application : NEWAPP           RC5100
Data Bases.....: 1           Pseudo Files : 0
Working Storage : 0           PCBs.....: 0

Screen                               Dynamic
--- Sync D.B. --- ----- Sync D.B. ----- Length
File Name  Qual  Method Path Call  Field..1  Field..2  Field
-----
PERSFILE   N    UPDATE
DEPTFILE   1
                DEPTNO
    
```

Synchronizing field 1

Press F3 twice to close the window and return to the Application Definition window.

```

VISION:Workbench for DOS           Working: NEWAPP
-----
RETRIEVE X=EXPORT PROTOTYPE
Application Name:NEWAPP           RC5000
Transact Application Sections
Files : Yes   Screen Flow: Yes   Environment: No   Comments: No
Screens : 4   Procs : 3   Requests : 0

Type  Name  Share  Description
-----
SCREEN MENU  NO    INITIAL MENU SCREEN
SCREEN NEWEMP NO    ENTER EMPLOYEE DATA
SCREEN SALINC NO    PROVIDE SALARY INCREASE DATA
SCREEN SALHIST NO    DISPLAY THE SALARY HISTORY
PROC  CHECKPF NO    CHECK THAT ONLY PF1, PF2 OR PF4 PRESSED
PROC  CHEKPF10 NO    CHECK THAT ONLY PF10 PRESSED
PROC  SALCALC NO    CALCULATE SALARIES FROM INCREASE DATA
    
```

CHOICES:Press F6 for choices; F8 to select this item.

Now, you need to write the NEWDATA procedure which calculates the new salary and the percentage of increase.

Position the cursor on the first blank line. Notice the instruction at the bottom of the screen.

Press F6 to open the Choices window.

VISION:Workbench for DOS Working: NEWAPP

RETRIEVE X=EXPORT PROTOTYPE

Application Name:NEWAPP RC5000

Transact Application Sections

Files : Yes Screen Flow: Yes Environment: No Comments: No
Screens : 4 Procs : 3 Requests : 0

Type	Name	Share	Description
SCREEN	MENU	NO	INITIAL MENU SCREEN
SCREEN	NEWEMP	NO	ENTER EMPLOYEE DATA
SCREEN			EASE DATA
SCREEN			HISTORY
PROC			, PF2 OR PF4 PRESSED
REQUEST			0 PRESSED
			FROM INCREASE DATA

Blank will select default - SCREEN.
Object will be used as a SCREEN.
Object will be used as a PROCEDURE.
Object will be used as a REQUEST.

Press ↑ ← Esc F3

Move the cursor to Proc and press Enter. (You could have just typed Proc instead of opening the Choices window.)

VISION:Workbench for DOS Working: NEWAPP

RETRIEVE X=EXPORT PROTOTYPE

Application Name:NEWAPP RC5000

Transact Application Sections

Files : Yes Screen Flow: Yes Environment: No Comments: No
Screens : 4 Procs : 3 Requests : 0

Type	Name	Share	Description
SCREEN	MENU	NO	INITIAL MENU SCREEN
SCREEN	NEWEMP	NO	ENTER EMPLOYEE DATA
SCREEN	SALINC	NO	PROVIDE SALARY INCREASE DATA
SCREEN	SALHIST	NO	DISPLAY THE SALARY HISTORY
PROC	CHECKPF	NO	CHECK THAT ONLY PF1, PF2 OR PF4 PRESSED
PROC	CHEKPF10	NO	CHECK THAT ONLY PF10 PRESSED
PROC	SALCALC	NO	CALCULATE SALARIES FROM INCREASE DATA
PROC	NEWDATA	NO	CALC NEW SALARY AND PERCENT INCREASE

Description of item; press F8 to select this item.

Give the procedure the name NEWDATA and add a description. (The Proc is assumed not to be shared with other applications. You could override this if you want.)

Press F8 to open the Procedure Definition window.

VISION:Workbench for DOS		Working:NEWAPP	
COPY MOVE SEARCH REPLACE			
Re-init Temps:		Procedure Name: NEWDATA	RQ5000
<pre> ; ; CALCULATE AND DISPLAY THE NEW SALARY AND THE PERCENTAGE INCREASE ; NEWSAL:FIELD P 4 2 ;TEMP FIELD TYPE=PACKED, LEN=4, DEC PLACES = 2 PCT: FIELD F 2 2 ;TEMP FIELD TYPE=FIXED, LEN=2, DEC PLACES = 2 ; LET T.NEWSAL = OLDSAL + AMOUNT ;NEW SALARY LET T.PCT = (AMOUNT/OLDSAL) * 100 ;PERCENT INCREASE ; TRANSFER TO OUTPUT ;OUTPUT THE LINE </pre>			
Enter a Procedure Statement			

Enter the procedure to calculate the new salary and percentage increase.

This is done for each line of the screen so the TRANSFER TO OUTPUT is necessary to display it.

Press F3 to close the Procedure Definition window.

You have written the procedure, but it has not yet been included in the application.

On the Application Definition window, move the cursor to Screen Flow:

- Press F8 to open the Screen Flow window.
- Move the cursor to SALHIST.
- Press F8 to open the Conditions window.

After OUTPUT SCREEN, add CALL NEWDATA.

VISION:Workbench for DOS		Working:NEWAPP	
PROTOTYPE EDIT			
Screen Flow for Application - NEWAPP			SC0100
<pre> graph TD MENU --> NEWEMP MENU --> SALINC MENU --> SALHIST </pre>			
Input Control Procedures for Screen : SALHIST Validate:CHERPF10 Retrieve: Update:			
_____ Output Conditions for SALHIST (from MENU) _____ IF F.ATTNID = 'PF004' OUTPUT SCREEN CALL NEWDATA END			
Free form OUTPUT conditions			

Proc NEWDATA will be activated for each of the lines of repeated data on the screen when the application is executed on the host. (This is the window you altered earlier. You could have added CALL NEWDATA when you changed PF003 to PF004. For purposes of illustration, changes are made one at a time.)

Press F3 to close the Conditions window.

Press F3 to close the Screen Flow window.

VISION:Workbench for DOS		Working:NEWAPP	
RETRIEVE X=EXPORT PROTOTYPE			
Application Name:NEWAPP			RC5000
Transact Application Sections			
Files	: Yes	Screen Flow:	Yes
Environment:	No	Comments:	No
Screens	: 4	Procs	: 4
Requests	: 0		
Type	Name	Share	Description
SCREEN	MENU	NO	INITIAL MENU SCREEN
SCREEN	NEWEMP	NO	ENTER EMPLOYEE DATA
SCREEN	SALINC	NO	PROVIDE SALARY INCREASE DATA
SCREEN	SALHIST	NO	DISPLAY THE SALARY HISTORY
PROC	CHEKPF	NO	CHECK THAT ONLY PF1, PF2 OR PF4 PRESSED
PROC	CHEKPF10	NO	CHECK THAT ONLY PF10 PRESSED
PROC	SALCALC	NO	CALCULATE SALARIES FROM INCREASE DATA
PROC	NEWDATA	NO	CALC NEW SALARY AND PERCENT INCREASE
Name of item: press F8 to select this item.			

Now you are ready to modify the screen.

Place the cursor on the SALHIST line and press F8 to open the Screen Structure window.

VISION:Workbench for DOS		Working:NEWAPP	
MOVE			
<input type="checkbox"/> Layout <input checked="" type="checkbox"/> Yes		Comments: 0	Screen Name:SALHIST SD0100
Input Transaction Code:		BMS/MFS Name....:	
Device ID.....:		Field Designator:	
Process Empty Segments:			
— Segment —	Occurs —	Display —	Virtual —
	Fields.....	
1	1	4	0
2	9	3	0
Required Name			

Position the cursor over Layout and press F8 to open the Layout window.

VISION:Workbench for DOS		Working:NEWAPP	
CENTER EDIT ATTRIBUTES X=CURSOR			
Line 1	Col 1	SCREEN LAYOUT - SALHIST	SD0200
REVIEW OF SALARY INCREASES FOR EMPLOYEE NUMBER: _____ FIRST NAME: _____ LAST NAME: _____ DEPT: <input type="checkbox"/> _____			
DATE	AMOUNT	OLD SALARY	
_____	_____	_____	
_____	_____	_____	
_____	_____	_____	
_____	_____	_____	
_____	_____	_____	
_____	_____	_____	
_____	_____	_____	
			PRESS PF10 TO RETURN TO THE MAIN MENU
Screen SALHIST; press F8 to define screen field.			

This is how the current SALHIST screen looks.

Position the cursor over the first of the four underscores of the Department Number field and press F8 to open the Display Specifications window.

VISION:Workbench for DOS		Working:NEWAPP	
COPY MOVE SEARCH REPLACE			
Line	4	Col	7
SCREEN LAYOUT - SALHIST			SD0200
REVIEW OF SALARY INCREASES FOR EMPLOYEE NUMBER: _____			
FIRST NAME: _____			
Segment.....:1	Name: DEPTNO	Type...:OUTPUT	Length.:
INPUT--->To..:		Action:	
Hilite Err...:	Num. Lock:	Display Modifications:	Protect:
OUTPUT-->From:	DEPTNO		
EITHER-->Num. Edit:		Edit Mask:	
— Additional Validation/Actions _____			
1-99			

Here you see that the screen field DEPTNO is an output field with the source of the data displayed obtained from the primary file field DEPTNO.

Position the cursor over the OUTPUT From field.

VISION:Workbench for DOS		Working:NEWAPP	
COPY MOVE SEARCH REPLACE			
Line	4	Col	7
SCREEN LAYOUT - SALHIST			SD0200
REVIEW OF SALARY INCREASES FOR EMPLOYEE NUMBER: _____			
FIRST NAME: _____			
Segment.....:1	Name: DEPTNO	Type...:OUTPUT	Length.:
INPUT--->To..:		Action:	
Hilite Err...:	Num. Lock:	Display Modifications:	Protect:
OUTPUT-->From:	1. DEPTDESC		
EITHER-->Num. Edit:		Edit Mask:	
— Additional Validation/Actions _____			
Name of source field			

Change the source of data to 1.DEPTDESC (the description from the DEPTFILE file).

Press F3 to close the window.

VISION:Workbench for DOS		Working:NEWAPP	
CENTER EDIT ATTRIBUTES X=CURSOR			
Line	4	Col	7
SCREEN LAYOUT - SALHIST			SD0200
REVIEW OF SALARY INCREASES FOR EMPLOYEE NUMBER: _____			
FIRST NAME: _____			
LAST NAME: _____			
DEPT: _____			
DATE	AMOUNT	OLD SALARY	
_____	_____	_____	<input type="checkbox"/>
_____	_____	_____	
_____	_____	_____	
_____	_____	_____	
_____	_____	_____	
_____	_____	_____	
_____	_____	_____	
_____	_____	_____	
			PRESS PF10 TO RETURN TO THE MAIN MENU
Screen SALHIST; press F8 to define screen field.			

Notice how the field designators for the DEPT field have changed to reflect the new source of data.

Position the cursor four spaces to the right of the OLD SALARY field on the first line of the repeated data.

Press F8 to open a Display Specifications window.

VISION:Workbench for DOS		Working:NEWAPP	
COPY MOVE SEARCH REPLACE			
Line	8	Col	50
SCREEN LAYOUT - SALHIST			SD0200
REVIEW OF SALARY INCREASES FOR EMPLOYEE NUMBER: _____			
FIRST NAME: _____			
Segment.....: 2	Name:NEWSAL	Type.:OUTPUT	Length.:8
INPUT--->To..:		Action:	
Hilite Err...:	Num. Lock:	Display Modifications:	Protect:
OUTPUT-->From:T.NEWSAL			
EITHER-->Num. Edit:	<input type="checkbox"/>	Edit Mask:	
— Additional Validation/Actions —			
Blank or 1-255			

Name the field on the screen NEWSAL, state that it is an OUTPUT field, and provide a display length of 8. Indicate the source of the data (the temporary field NEWSAL) and that numeric editing is wanted.

Press F3 to close the Display Specifications window.

Move the cursor four spaces to the right of the new field designator and press F8 to open a new Display Specifications window.

VISION:Workbench for DOS		Working:NEWAPP	
COPY MOVE SEARCH REPLACE			
Line	8	Col 62	SCREEN LAYOUT - SALHIST SD0200
REVIEW OF SALARY INCREASES FOR EMPLOYEE NUMBER: _____ FIRST NAME: _____			
Segment.....:	2	Name:PERCENT	Type.:OUTPUT Length.:6
INPUT--->To..:			Action:
Hilite Err...:		Num. Lock:	Display Modifications: Protect:
OUTPUT-->From:	T.PCT		
EITHER-->Num. Edit:			Edit Mask:
— Additional Validation/Actions —			
Name of source field			

Provide the specifications of the Percent field.

Screen field name: PERCENT Type: OUTPUT
Length: 6 Output from: T.PCT

Press F3 to close the window.

With the two fields in place, you can, optionally, type in column headings. Use the arrow keys to position the cursor to where you want to enter the column headings.

VISION:Workbench for DOS		Working:NEWAPP	
CENTER	EDIT	ATTRIBUTES	X-CURSOR
Line 7	Col 69	SCREEN LAYOUT - SALHIST	SD0200
REVIEW OF SALARY INCREASES FOR EMPLOYEE NUMBER: _____ FIRST NAME: _____ LAST NAME: _____ DEPT: _____			
DATE	AMOUNT	OLD SALARY	NEW SALARY PERCENT INCREASE
_____	_____	_____	_____
_____	_____	_____	_____
_____	_____	_____	_____
_____	_____	_____	_____
_____	_____	_____	_____
_____	_____	_____	_____
_____	_____	_____	_____
PRESS PF10 TO RETURN TO THE MAIN MENU			
Perform editing functions			

To complete the columns, press F4 to move the cursor to the menu bar. Position the cursor over EDIT and press Enter.

VISION:Workbench for DOS		Working:NEWAPP	
EXCHANGE	COPY	MOVE	SEARCH REPLACE
Line 7	Col 69	SCREEN LAYOUT - SALHIST	SD0200
REVIEW OF SALARY INCREASES FOR EMPLOYEE NUMBER: _____ FIRST NAME: _____ LAST NAME: _____ DEPT: _____			
DATE	AMOUNT	OLD SALARY	NEW SALARY PERCENT INCREASE
_____	_____	_____	_____
_____	_____	_____	_____
_____	_____	_____	_____
_____	_____	_____	_____
_____	_____	_____	_____
_____	_____	_____	_____
_____	_____	_____	_____
PRESS PF10 TO RETURN TO THE MAIN MENU			
Copy rectangle			

A new menu bar appears. Select COPY.

Position the cursor to the top left of the data to be copied and press Enter. (In this case, this is just in front of the two new fields you defined.)

VISION:Workbench for DOS		Working:NEWAPP			
CENTER EDIT ATTRIBUTES X=CURSOR					
Line	8	Col	49	SCREEN LAYOUT - SALHIST	SD0200
REVIEW OF SALARY INCREASES FOR EMPLOYEE NUMBER: _____					
FIRST NAME: _____					
LAST NAME: _____					
DEPT: _____					
DATE	AMOUNT	OLD SALARY	NEW SALARY	PERCENT INCREASE	
_____	_____	_____	_____	_____	
_____	_____	_____	_____	_____	
_____	_____	_____	_____	_____	
_____	_____	_____	_____	_____	
_____	_____	_____	_____	_____	
_____	_____	_____	_____	_____	
_____	_____	_____	_____	_____	
PRESS PF10 TO RETURN TO THE MAIN MENU					
Position cursor to lower right of area to be moved; then ↵					

A small box will open up.
Follow the instructions at the bottom of the screen.

VISION:Workbench for DOS		Working:NEWAPP			
CENTER EDIT ATTRIBUTES X=CURSOR					
Line	8	Col	69	SCREEN LAYOUT - SALHIST	SD0200
REVIEW OF SALARY INCREASES FOR EMPLOYEE NUMBER: _____					
FIRST NAME: _____					
LAST NAME: _____					
DEPT: _____					
DATE	AMOUNT	OLD SALARY	NEW	PERCENT	
_____	_____	_____	_____	_____	
_____	_____	_____	_____	_____	
_____	_____	_____	_____	_____	
_____	_____	_____	_____	_____	
_____	_____	_____	_____	_____	
_____	_____	_____	_____	_____	
_____	_____	_____	_____	_____	
PRESS PF10 TO RETURN TO THE MAIN MENU					
Position cursor to lower right of area to be moved; then ↵					

Press Enter when you have surrounded the data to copy.

VISION:Workbench for DOS		Working:NEWAPP	
CENTER	EDIT	ATTRIBUTES	X=CURSOR
Line 8	Col 49	SCREEN LAYOUT - SALHIST	SD0200
REVIEW OF SALARY INCREASES FOR EMPLOYEE NUMBER: _____			
FIRST NAME: _____			
LAST NAME: _____			
DEPT: _____			
DATE	AMOUNT	OLD SALARY	NEW PERCENT
_____	_____	_____	<input type="text"/>
_____	_____	_____	_____
_____	_____	_____	_____
_____	_____	_____	_____
_____	_____	_____	_____
_____	_____	_____	_____
_____	_____	_____	<input type="text"/>
			THE
MAIN MENU			
Position cursor to destination, then ↓ ; use Esc to quit			

Press the down arrow and the Enter key eight times.

Each press of Enter will copy the source box. Each down arrow will move the target box.

Press Esc at the end.

The screen is now completely designed.

VISION:Workbench for DOS		Working:NEWAPP		
CENTER	EDIT	ATTRIBUTES	X=CURSOR	
Line 16	Col 68	SCREEN LAYOUT - SALHIST	SD0200	
REVIEW OF SALARY INCREASES FOR EMPLOYEE NUMBER: _____				
FIRST NAME: _____				
LAST NAME: _____				
DEPT: _____				
DATE	AMOUNT	OLD SALARY	NEW SALARY	PERCENT INCREASE
_____	_____	_____	_____	_____
_____	_____	_____	_____	_____
_____	_____	_____	_____	_____
_____	_____	_____	_____	_____
_____	_____	_____	_____	_____
_____	_____	_____	_____	_____
_____	_____	_____	_____	_____
				PRESS PF10 TO RETURN TO THE MAIN MENU
Change display attributes				

The last task is to color the percent column red.

Press F4 to move the cursor to the menu bar.

Position the cursor over the ATTRIBUTES and press Enter.

VISION:Workbench for DOS		Working:NEWAPP		
PAINT				
Line 16	Col 68	SCREEN LAYOUT - SALHIST	SD0200	
REVIEW OF SALARY INCREASES FOR EMPLOYEE NUMBER: _____				
FIRST NAME: _____				
LAST NAME: _____				
DEPT: _____				
DATE	AMOUNT	OLD SALARY	NEW SALARY	PERCENT INCREASE
_____	_____	_____	_____	_____
_____	_____	_____	_____	_____
_____	_____	_____	_____	_____
_____	_____	_____	_____	_____
_____	_____	_____	_____	_____
_____	_____	_____	_____	_____
_____	_____	_____	_____	_____
				PRESS PF10 TO RETURN TO THE MAIN MENU
CHOICES: <blank>, NEUTRAL, DEFAULT, BLUE, RED, PINK, GREEN, TURQUOISE, YELLOW				

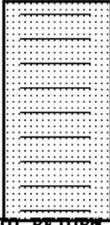
Change Display Attributes

Color: **RED** Intensity: _____

Hilite: _____ Pgm Symbol: _____

The Display Attributes window appears. You could press F6 to get the list of possible colors, but because you know you want red, just type RED. (When you type R, the remainder of the word is automatically provided.)

Press F4 to move the cursor to the menu bar. Select PAINT. Position the cursor in front of the first field of the Percent Increase column and press Enter. Notice that the cursor changed to red (unless you have a monochrome screen).

VISION:Workbench for DOS		Working:NEWAPP		
PAINT				
Line 16	Col 68	SCREEN LAYOUT - SALHIST		SD0200
REVIEW OF SALARY INCREASES FOR EMPLOYEE NUMBER: _____				
FIRST NAME: _____				
LAST NAME: _____				
DEPT: _____				
DATE	AMOUNT	OLD SALARY	NEW SALARY	PERCENT INCREASE
_____	_____	_____	_____	_____
_____	_____	_____	_____	_____
_____	_____	_____	_____	_____
_____	_____	_____	_____	_____
_____	_____	_____	_____	_____
_____	_____	_____	_____	_____
_____	_____	_____	_____	_____
_____	_____	_____	_____	_____
				
				PRESS PF10 TO RETURN TO THE MAIN MENU
Position cursor to lower right of area to be painted; then ↵				

To paint the column, move the right arrow to the right side of the first Percent Increase field and move the down arrow to the bottom right of the last Percent Increase field. The column underscores change to red. (With monochrome terminals, the column becomes reverse video.)

Press Enter when you have painted the column.

What you see is what will be displayed on a 3270 terminal when the application is actually executed, except that underscores are not displayed. Underscores are replaced by output data for those segments which actually exist.

Press F3 twice to close the Display Attributes window and the Layout window.

Press F3 to close the Screen Structure window and return to the Main Menu.

VISION:Workbench for DOS		Working:NEWAPP	
RETRIEVE <input checked="" type="checkbox"/> EXPORT PROTOTYPE			
Application Name:NEWAPP			RC5000
Transact Application Sections			
Files	: Yes	Screen Flow:	Yes Environment: No Comments: No
Screens	: 4	Procs	: 4 Requests : 0
Type	Name	Share	Description
SCREEN	MENU	NO	INITIAL MENU SCREEN
SCREEN	NEWEMP	NO	ENTER EMPLOYEE DATA
SCREEN	SALINC	NO	PROVIDE SALARY INCREASE DATA
SCREEN	SALHIST	NO	DISPLAY THE SALARY HISTORY
PROC	CHEKPF	NO	CHECK THAT ONLY PF1, PF2 OR PF4 PRESSED
PROC	CHEKPF10	NO	CHECK THAT ONLY PF10 PRESSED
PROC	SALCALC	NO	CALCULATE SALARIES FROM INCREASE DATA
PROC	NEWDATA	NO	CALC NEW SALARY AND PERCENT INCREASE
F1	F2	F3	F4
Help	Names	Close	Menu Bar
F5	F6	F7	F8
Save	Choices	Validate	Open
F9	Alt+F10		
Insert	Delete		

Now, you have finished the modifications.

Press F1 to display the function key bar.

This reminds you that F7 will validate the application.

Press F7.

VISION:Workbench for DOS		Working:NEWAPP	
RETRIEVE <input checked="" type="checkbox"/> EXPORT PROTOTYPE			
Application Name:NEWAPP			RC5000
Transact Application Sections			
Files	: Yes	Screen Flow:	Yes Environment: No Comments: No
Screens	: 4	Procs	: 4 Requests : 0
Type	Name	Share	Description
SCREEN	MENU	NO	INITIAL MENU SCREEN
SCREEN	NEWEMP	NO	ENTER EMPLOYEE DATA
SCREEN	SALINC	NO	PROVIDE SALARY INCREASE DATA
SCREEN	SALHIST	NO	DISPLAY THE SALARY HISTORY
PROC	CHEKPF	NO	CHECK THAT ONLY PF1, PF2 OR PF4 PRESSED
PROC	CHEKPF10	NO	CHECK THAT ONLY PF10 PRESSED
PROC	SALCALC	NO	CALCULATE SALARIES FROM INCREASE DATA
PROC	NEWDATA	NO	CALC NEW SALARY AND PERCENT INCREASE
Export the current application			

Press F5 to save the application.

Now it is time to export the application to the host. Currently the application is in an internal format only readable by VISION:Workbench. In order to process the application by VISION:Transact, this internal format must first be converted to a format readable by VISION:Transact. This conversion is done by EXPORT.

EXPORT creates 80-position images that you can add job control to and send to the host for processing by VISION:Transact as M5INPUT.

Press F4 to move the cursor to the menu bar. Select EXPORT.

(You could of course select PROTOTYPE to prototype the application and verify that PF4 is active in place of PF3.)

VISION:Transact does not send anything to the host. It generates 80-position input statements into two PC-DOS files that can be subsequently shipped to the host.

The files are given the name of the application with a suffix appended to distinguish their contents. One of the files has .DEF appended; this file contains definition statements for the application. The other file has .GEN appended; this file contains procedure and report statements. The files generated by this application are named NEWAPP.GEN and NEWAPP.DEF. You can use your PC-host communication program to transfer these files to the host for processing.

Press F3 twice to return to the Main Menu.

Utilities

VISION:Workbench utilities allow you to print applications and to import applications and definitions that have been downloaded from the host. You can also temporarily override the Import Input Path and Printer Output Destination previously entered in the Setup window. You select these utilities on the Main Menu by selecting the Utilities entry in the Miscellaneous box. This opens the Utilities window shown in [Figure 12-1](#).

VISION:Workbench for DOS										Working:EXAMPLE									
BUILDER 0 = TRANSACT FILE X=TRANSACTION ARRAY TABLE																			
Select an activity :										UTIL									
<table border="1"> <thead> <tr> <th colspan="2">Utility Functions</th> </tr> </thead> <tbody> <tr> <td>Application Import Facility</td> <td></td> </tr> <tr> <td>Definition Import Facility</td> <td></td> </tr> <tr> <td>Print Library Items</td> <td></td> </tr> </tbody> </table>												Utility Functions		Application Import Facility		Definition Import Facility		Print Library Items	
Utility Functions																			
Application Import Facility																			
Definition Import Facility																			
Print Library Items																			
<table border="1"> <thead> <tr> <th colspan="2">Override Pathnames</th> </tr> </thead> <tbody> <tr> <td>Import Input Path:</td> <td>C:\MPCXX\IMPORT</td> </tr> <tr> <td>Printer Output Dest:</td> <td>PRN</td> </tr> </tbody> </table>												Override Pathnames		Import Input Path:	C:\MPCXX\IMPORT	Printer Output Dest:	PRN		
Override Pathnames																			
Import Input Path:	C:\MPCXX\IMPORT																		
Printer Output Dest:	PRN																		
F1 Help	F2	F3 Close	F4	F5	F6	F7	F8	F9	Alt+F10										
Press ENTER or F8 to select this item																			

Figure 12-1 Utilities Window

Print Library Items Utility

The Print Library Items utility can be used to print all the parts of an application or portions of applications, such as file definitions only or procedures only.

You select the print utility by moving the cursor to Print Library Items and pressing F8. The following items appear on the menu bar at the top of the Utility window and can be selected for printing:

- Builder
- Transaction Group
- Transact
- Array
- File
- Table

Selecting Builder displays the menu bar items (Subfile(s), Report(s), Application(s), Procedure(s), Request(s)) for printing those parts of VISION:Builder applications. Selecting any of these results in the display of the All, Selections, and Names menu bar items. These menu items are described at the end of this section.

- Subfile(s) prints Subfile Definition windows.
- Application(s) prints Application Definition windows.
- Report(s) prints Report Definition windows.
- Procedure(s) prints Procedure Definition windows.
- Request(s) prints Request Definition windows.

Selecting Transact displays the menu bar items (Screen(s), Application(s), Procedure(s), Request(s)) for printing those parts of VISION:Transact applications. Selecting any of these results in the display of the All, Selections, and Names menu bar items. These menu items are described at the end of this section.

- Screen(s) prints Screen Definition windows.
- Application(s) prints Application Definition windows.
- Procedure(s) prints Procedure Definition windows.
- Request(s) prints Screen Processing Procedure windows.

Selecting FILE, TRANSACTION, ARRAY, or TABLE opens the window that lists the names of selected item type. The ALL, SELECTIONS, and NAMES menu bar items appear on this window. These menu items are described in a later section.

VISION:Workbench for DOS		Working:EXAMPLE	
ALL SELECTIONS NAMES			
Builder Applications on Library			MK0200
EX1	EXAMPLE	OLYMPICS	TRANRUN
Press ← to select item(s) to print, [Esc] to end			

Figure 12-2 Print Selection Menu Bar

ALL, SELECTIONS, and NAMES Menu Bar Items

ALL (see [Figure 12-2](#)) prints the contents of all the item names appearing on the list window. You display this menu bar by pressing Esc and either pressing the letter A or moving the cursor to ALL and pressing Enter.

SELECTIONS (see [Figure 12-2](#)) prints the contents of only selected item names appearing on the list window. You use this menu item as follows:

1. To select item names for printing, move the cursor to the item name whose contents you want to print and press Enter. Move the cursor off the name. Now it is highlighted for printing. Select as many names as you like in this manner.
2. When you are ready to print, press Esc. The cursor moves to the menu bar.
3. To select this item, either press the letter S or move the cursor to SELECTIONS and press Enter. The items that you selected begin printing.

NAMES (see [Figure 12-2](#)) prints all the names appearing on the list window. You display this menu bar by pressing Esc and either pressing the letter N or moving the cursor to NAMES and pressing Enter.

Importing Applications and Definitions

VISION:Workbench has an extremely useful built-in conversion utility for importing your existing host definitions and applications to the VISION:Workbench. You can take any of your VISION:Builder or VISION:Transact applications or definitions on the host, download them to the PC, and tell VISION:Workbench their PC-DOS file name. VISION:Workbench converts them into the VISION:Workbench format. Once conversion is complete, the definitions and applications are stored in the VISION:Workbench library and they appear in their appropriate display list.

This utility saves you time of reentering definitions and applications that already work on the host while eliminating the probability of errors during reentering.

The conversion utility assigns the definition a PC-DOS file name and suffix. The suffix is determined by the definition type:

.FDN	File Definitions
.ARD	Array Definitions
.TAB	Table Definitions
.TRG	Transaction Definitions
.RC4	VISION:Builder Application Definitions
.RC5	VISION:Transact Application Definitions
.SN5	Screen Definitions

The PC-DOS file name (minus the suffix) and VISION:Workbench definition name are derived from the file name in positions 1 through 8 of the original VISION:Builder source statement. For example, the file name BDBCUSTE.FDN is created from the following VISION:Builder source statements:

```
BDBCUSTEFDBBCUST          BD      05000
BDBCUSTELSCUSTOMER        01
BDBCUSTEL0CUSTNO          0110001005C1  05
BDBCUSTEL0CUSTNAME        0110006030C   30
BDBCUSTEL0CUSTPH          0110036010C   10
```

The FD in positions 9 and 10 of the first statement is used to determine the suffix.

There are a few requirements you must satisfy before you use the conversion utility. Some must be performed on the host, some on the PC. The following describes the host requirements:

1. The definitions that you want to download must be processed by source statement retrieval and put into 80-position statements. (See the VISION:Builder or VISION:Transact Specifications Guide for detailed information about source statement retrieval.) Applications that you want to download must be complete in a form ready for host execution.
2. Put the definitions and applications into a file that can be sent through your PC to host file transfer program. This may require you to convert each file to ASCII before transmitting it. See your PC to host file transfer program documentation for specific requirements.
3. Download the file(s) to the PC.

The following describes the steps you must take before you can convert definitions on the PC:

1. Copy the files into your application or importing subdirectory.
2. Be sure to assign the files a suffix of “.DEF” for definitions, “.RUN” for VISION:Builder applications, and “.GEN” for VISION:Transact applications.

Now you are ready to use the import utility. Begin by getting into the VISION:Workbench root directory and starting the VISION:Workbench. Type:

```
cd \mpcxx (where xx is the release number)
mpcx
```

To use the import facility, select Utilities from the main menu. The Utilities window is shown again in [Figure 12-3](#).

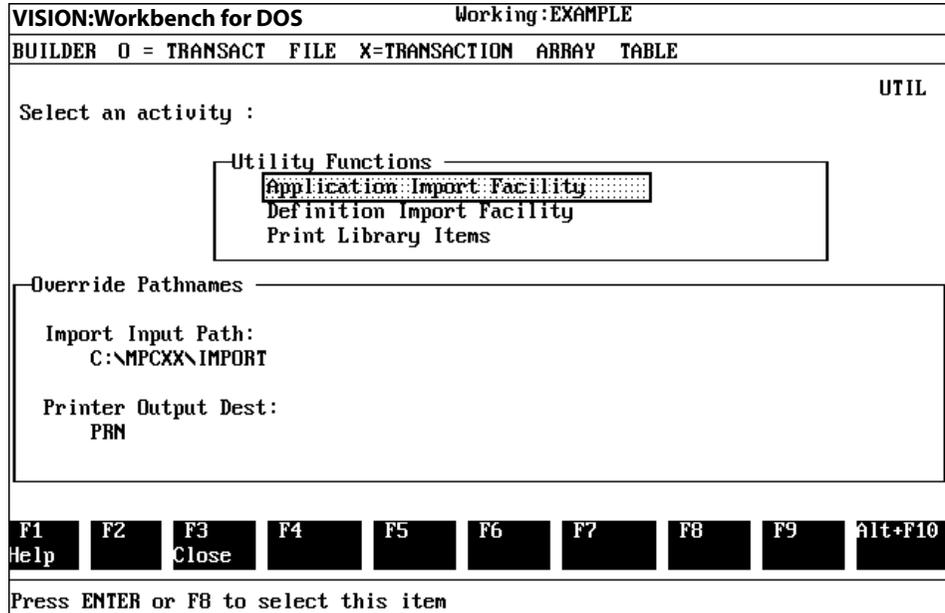


Figure 12-3 Utilities Window

From this window, select either the Application Import Facility or Definition Import Facility depending on what type of item you want to import. The files you import must be located in the Import Input Path. The lower window shows the default paths you provided during the setup process. The import path identifies where the source to import is located; the printer output destination identifies where to send any diagnostics that result from the import. You can temporarily override these paths from this window.

After selecting the appropriate import facility, a list of file names from the current Import Input Path is displayed. When you select the Application Import Facility, you must select whether you want VISION:Builder or VISION:Transact applications imported. All appropriate files with a .RUN or .GEN extension are displayed. If you select the Definition Import Facility, all files with a .DEF extension are displayed.

Figure 12-4 shows the resulting list from which you select the item you want to import.

VISION:Workbench for DOS	Working:EXAMPLE
BUILDER 0 = TRANSACT ONLINE FILE X=TRANSACTION ARRAY TABLE	
Select an activity :	UTIL
<div style="border: 1px solid black; padding: 5px; margin: 10px auto; width: fit-content;"> Utility Functions Application Import Facility Definition Import Facility Print Library Items </div>	
Override Pathnames Import Input Path: C:\MPCXX\IMPORT Printer Output Dest: PRN	
↑ * ← Esc F3	
EXAMPLE MSGUPDAT OLYMPICS	
Select item to import with arrow keys; then ↓	

Figure 12-4 Select Item to Import

After selecting the appropriate file name, a prevalidation scan for host features that are not supported by the Workbench is done on the file. If any unsupported features are found, a listing is sent to the printer output destination and a summary window is displayed. A sample listing and summary window are shown in [Figure 12-5](#) and [Figure 12-6](#).

```
VISION:Workbench for DOS

                                IMPORT VALIDATION SUMMARY  -  01-15-95
                                -----

Object XXXXXXXX : Sort control not supported
Object XXXXXXXX : Override delimiter not supported
Object XXXXXXXX : Spool blocking factor not supported
Object XXXXXXXX : Library maintenance not supported
Object XXXXXXXX : Item tracking not supported

Object MGNTRAM1 : Mapped files not supported

STATEMENTS VALIDATED : 763
MESSAGES   ISSUED    : 6
```

Figure 12-5 Sample Listing of Unsupported

```
Current:

The following functions are not supported :

Sort control  Override delim.  Spoolblocking  Library maint.
Item tracking  Mapped files

If the import proceeds these statements will be removed from the
application.

Continue :  YES  NO

Press enter to select this item
```

Figure 12-6 Sample Summary Window

When the Summary window is displayed, you have the option of continuing or terminating the import process. If you choose to continue with the import of the file, the unsupported features are removed from the application or definition.

If the import process is continued, the application or definition is converted into a VISION:Workbench internal format and is saved in the current Primary library (as specified on the Setup window). If an application or definition with the same name already exists in the library, you have the option of replacing it.

When the import process is complete, you are returned to the Main Menu.

Transaction Definition Import Considerations

On the host, a transaction definition is a combination of file definition and action specification. In VISION:Workbench, a transaction is a predefined interaction between two files. Therefore, transactions that are imported into VISION:Workbench must be restructured.

When importing transaction definitions, the following window, shown in [Figure 12-7](#), is opened.

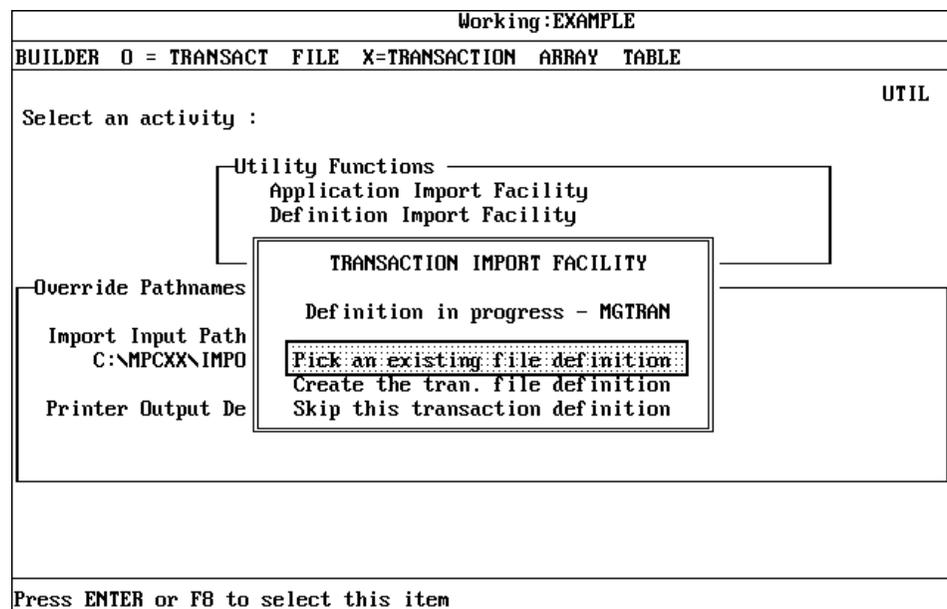


Figure 12-7 Transaction Import Window

From this window, there are two ways of specifying the transaction file definition to be used or you can choose not to import the transaction.

If you select the first option, Pick an existing file definition, a list of existing file definitions is displayed. From the list, you can choose the file definition which corresponds to your transaction file format to be used in the new transaction definition.

If you select the second option, Create the transaction file definition, you are prompted to supply a name for the definition. VISION:Workbench automatically creates the transaction file definition for you.

Application Import Considerations

VISION:Workbench for DOS applications are constructed by gathering together different objects, such as control procedures, reports, and subfiles. In contrast to VISION:Builder and VISION:Transact, in VISION:Workbench, the objects are distinct and separate. Therefore, applications that are imported into VISION:Workbench must be restructured to conform to the system's structured application convention. This process is performed automatically by VISION:Workbench.

In general, an application that is imported into VISION:Workbench does not have event controlled procedures. VISION:Workbench automatically creates event controlled procedures, changes some requests to subroutine requests, and, sometimes, inserts new statements into existing requests. Each report and subfile is placed into its unique subroutine and calls to these subroutines are inserted at appropriate points to maintain the integrity of your application. While the source of the application is significantly changed, the integrity of the application is preserved (with the exception of any features you permit to be deleted in the prevalidation step).

Exporting to the Host

The Export facility of VISION:Workbench creates PC-DOS files that can be sent to your host for execution by VISION:Builder or VISION:Transact. When you select the EXPORT menu bar item from the Application Definition window, the Workbench creates two PC-DOS files. It generates 80-position statements into each file that define specific parts of the application. The statements can be easily identified by looking at the suffix that VISION:Workbench assigns to each file name.

File names appended with the “.DEF” suffix contain the application's definition statements. These statements describe the structure and contents of files used in the application.

File names appended with either a “.RUN” or “.GEN” suffix contain statements that describe the VISION:Builder or VISION:Transact application, respectively: input/output files, procedural logic, and reports.

You can ship these files to your host with any PC to host communication program. If you want to keep a permanent copy of your application, you must SAVE it either before or after you do the EXPORT. Save it by pressing F5 (SAVE) on the Application Definition window.

What EXPORT Does

The following describes the steps that take place when you ask VISION:Workbench to do an EXPORT:

1. If the application or definition has not been previously validated, VISION:Workbench asks if you want to select or suppress prevalidation. If the validation is suppressed, the application or definition may be EXPORTed with errors. This is useful when you want to integrate part of a Workbench application with an existing application on the host.
2. For VISION:Builder applications, a pop-up window appears which allows you to select the type of host processing you will want when the application is executed on the host. Following are the choices and a description of each:

Normal Execution	Select Normal Execution for all valid procedures and requests to be processed.
Sample Reports Only	Select Sample Reports Only if you want a sample report generated for each valid procedure and request in the run. Sample reports allow you to check the appearance of your data without actually preparing test data.
Syntax Validation Only	Select Syntax Validation Only if you want the procedures and requests syntax checked, cataloged request maintenance performed, and diagnostics issued. No processing is done.

3. Applications and definitions are generated into 80-column statements.
4. Definitions are generated into a PC-DOS file that uses the naming convention XYZ.DEF, where XYZ is the name of your definition or application and DEF is the suffix that VISION:Workbench assigns to definition files.
5. The other parts of your application (input/output file names, procedures, reports, and subfiles) are generated into another PC-DOS file that uses the naming convention XYZ.RUN or XYZ.GEN, where XYZ is the name of your application and RUN or GEN is the suffix that VISION:Workbench assigns to VISION:Builder or VISION:Transact application files, respectively.

About EXPORTing Files

Both a .DEF and a .RUN or .GEN file are generated when you export an application. It is recommended that you use the .DEF file in a dictionary maintenance run, placing the definition into a temporary COMLIB. This temporary COMLIB should be used while executing the application from your .RUN or .GEN file. This way you ensure that your host application is using the identical definitions used by VISION:Workbench for your application development.

VISION:Workbench creates the .DEF, .RUN, and .GEN files in the EXPORT output path directory that you specify in the SETUP information. The directory must be created in DOS and defined in the Setup window before you do the export. If it is not defined, VISION:Workbench uses the application subdirectory instead. (See [Specifying Setup Information on page 2-4](#) for detailed information on how to define the submission directory.)

VISION:Workbench uses the definition or application name to name the PC-DOS file containing the definitions (.DEF) and other VISION:Builder application parts (.RUN) or VISION:Transact application parts (.GEN). If you make changes to your definition or application and re-export it without changing its name, VISION:Workbench writes over the previously exported files. This is fine if you are not planning to save the .DEF, .RUN, and .GEN files. If you want to save them and create new files as well, be sure to change the name of the definition or application before exporting it again or use the DOS RENAME command to rename the original versions.

A VISION:Builder User's Guide to VISION:Workbench

This appendix is intended for existing VISION:Builder users. It attempts to explain VISION:Workbench VISION:Builder Applications in VISION:Builder fixed format terms. Features for which the correspondence between the two products is fairly obvious are not included. This appendix is not intended as a substitute for reading the rest of this guide. It merely attempts to point you in the right direction in those situations where the correspondence between VISION:Builder and VISION:Workbench may not be immediately apparent.

Most of VISION:Workbench features mentioned here are discussed in greater detail in [Chapter 6, *Creating VISION:Builder Applications*](#) or [Chapter 7, *Creating Reports*](#) of this guide. The ASL language is described in the ASL Reference Guide.

Naming Conventions

Except in requests, VISION:Workbench requires qualified field names to be entered as a single entity: the qualifier, followed by a period (.) which is followed by the field name (for example, 1.CUSTNO). Unqualified names are the same as in VISION:Builder (for example, CUSTNO).

Partial fields are represented by an ASL PF function. For example, PF(CUSTOMER,3,2) specifies a partial field of CUSTOMER starting at position 3 for a length of 2.

Comment (AA) Statement

TEXT – comments can be specified in several places within a VISION:Workbench VISION:Builder application:

- The Comments Section associated with the Application Definition window.
- The Comments Section associated with the File Definition window.
- The Comments Section associated with the Request Definition window.
- On any ASL statement, following a semicolon (;).

Establish Request (ER) Statement

VISION:Workbench introduces a new entity (the procedure) which is equivalent to a request. (Requests are still supported, however.) Procedures use a free-form, structured language called ASL (Advanced Syntax Language).

CATALOG ACTION – not applicable.

REQUEST TYPE – not used. VISION:Workbench infers the type from the way in which a request (or procedure) is used:

- Type 1, 2, or M requests are identified by referencing them in an Event Controlled Procedures window as “Tran-Record Initial Validation”, “Tran File and Master File Aligned”, and “After Update, Transaction View” input control procedures, respectively. Only one request/procedure of a given type can be specified for any screen. However, if, for example, you require several type 1 procedures, simply specify a single one as the Validate procedure for the screen and call the rest from that procedure.
- Type S requests and procedures are identified by calling them from within a procedure.
- Type E requests are identified by referencing them in the Event Controlled Procedures window as “End of File”.
- Type 3 and 4 requests are identified by referencing them in the Event Controlled Procedures window as “After Update, Entire-Record View” and “Transaction Error Handling,” respectively.

BACK BRANCH LIMIT – supported in requests. In procedures, you can use a DO/END block to accomplish the same thing.

Procedural (PR) Statement

PR statements in Workbench requests are identical to those in VISION:Builder, except that the CALL operator is not supported. However, the ASL language used in procedures supports all of the VISION:Builder PR statement functions including the CALL feature.

Run Control (RC) Statement

MESSAGE PROCESSING OPTIMIZATION – stored internally and exported.

OVERRIDE DELIMITER – not used. The system delimiter is specified in the System Parameters window (see [Chapter 2, Getting Started – Installation and Setup](#)).

REPORT FILE OPTIMIZATION – R is supported. S is not supported.

RUN NAME – specified as the Application Name in the Application Definition window.

SOURCE STATEMENT LISTING – stored internally and exported.

SOURCE STATEMENT OUT – not applicable.

SPOOL BLOCKING FACTOR – not applicable.

STORAGE OPTIMIZATION – not applicable.

Application Files (RF) Statement

Files used within an application are specified with the Files Used window.

Run Parameter (RP) Statement

Most of the entries for the RP statement are entered on the Application Parameters window. However, the CHKORDER option is entered on the Files Used window.

Temporary Field (TF) Statement

Temporary Fields can be specified on the Temporary Field window (which is accessed from the Request Definition window) or by entering FIELD statements within the Procedure Definition window.

A VISION:Transact User's Guide to VISION:Workbench

This appendix is intended for existing VISION:Transact users. It attempts to explain VISION:Workbench for DOS VISION:Transact Applications in VISION:Transact fixed format terms. Features for which the correspondence between the two products is fairly obvious are not included. This appendix is not intended as a substitute for reading the rest of this guide. It merely attempts to point you in the right direction in those situations where the correspondence between VISION:Transact fixed format and VISION:Workbench may not be immediately apparent.

Most of VISION:Workbench features mentioned here are discussed in greater detail in [Chapter 9, *Creating VISION:Transact Applications*](#) or [Chapter 10, *Screen Layout, Contents, and Actions*](#) of this guide. The ASL language is described in the ASL Reference Guide.

Naming Conventions

Except in requests, VISION:Workbench requires qualified field names to be entered as a single entity: the qualifier, followed by a period (.) which is followed by the field name (for example, 1.CUSTNO). Unqualified names are the same as in VISION:Transact (for example, CUSTNO).

Partial fields are represented by an ASL PF function. For example, PF(CUSTOMER,3,2) specifies a partial field of CUSTOMER starting at position 3 for a length of 2.

Named patterns are specified using a P qualifier (for example, P.NUMERIC).

Comment (AA) Statement

TEXT – comments can be specified in several places within a VISION:Workbench VISION:Transact application:

- The Comments Section associated with the Application Definition window.
- The Comments Section associated with the Screen Definition window.
- The Comments Section associated with the Request Definition window.
- On any ASL statement, following a semicolon (;).

Establish Request (ER) Statement

VISION:Workbench introduces a new entity (the procedure) which is equivalent to a request. (Requests are still supported, however.) Procedures use a free-form, structured language called ASL (Advanced Syntax Language).

CATALOG ACTION – not applicable.

REQUEST TYPE – not used. VISION:Workbench infers the type from the way in which a request (or procedure) is used:

- Type 1, 2, or M requests are identified by referencing them in a Screen Flow Conditions window as Validate, Retrieve, or Update input control procedures, respectively. Only one request/procedure of a given type can be specified for any screen. However, if, for example, you require several type 1 procedures, simply specify a single one as the Validate procedure for the screen and call the rest from that procedure.
- Type O requests and procedures are identified by naming them as the CALL operand of an OUTPUT statement within a Screen Flow Conditions window.
- Type S requests and procedures are identified by calling them from within a procedure.
- Type N requests are identified simply by the fact that they contain only temporary field definitions. Type N procedures are identified by the fact that they contain only FIELD statements (the ASL equivalent of temporary field definitions).

SCREEN NAME – not used. VISION:Workbench infers the screen/request relationship in the same manner as described above for REQUEST TYPE.

BACK BRANCH LIMIT – supported in requests. In procedures, you can use a DO/END block to accomplish the same thing.

Procedural (PR) Statement

PR statements in VISION:Workbench requests are identical to those in VISION:Transact, except that the CALL operator is not supported. However, the ASL language used in procedures supports all of the VISION:Transact PR statement functions including the CALL feature.

Run Control (RC) Statement

Most of these specifications are made using the Application Environment window, except for the following:

RUN NAME – specified as the Application Name in the Application Definition window.

RUN TYPE – not used. The selection you choose from VISION:Workbench Main Menu determines this.

OVERRIDE DELIMITER – not used. The system delimiter is specified in the System Parameters window (see [Chapter 2, Getting Started – Installation and Setup](#)).

SOURCE STATEMENT LISTING – not applicable.

Application Files (RF) Statement

Files used within an application are specified with the Files Used window. This window has four subordinate windows, one for each type of file — database, pseudo file, working storage, and PCB.

FILE TYPE – not used. This is inferred from the type of subordinate window you use.

SYNC QUALIFIER – must be specified as part of the synchronizing field (for example, 1.CUSTNO).

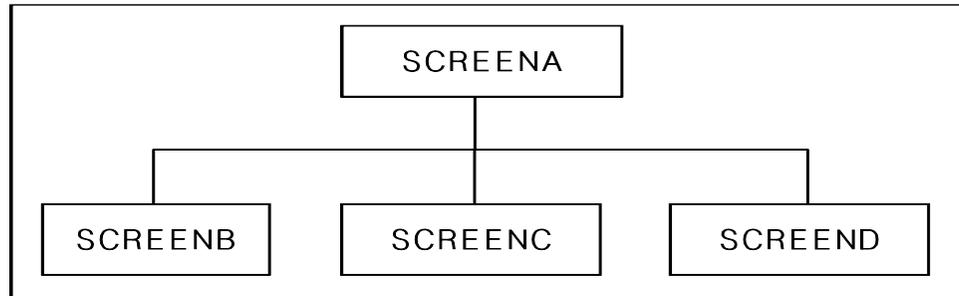
SCREEN SYNC TYPE – note that P (for pseudo file) is not used. You indicate that a file is a pseudo file by choosing the appropriate subordinate window.

Run Parameter (RP) Statement

PARAMETER NAME – only one RP feature, FASTPATH, is supported by VISION:Workbench. This specification is made in the Application Environment window.

Screen Control (SC) Statement

The relationship between input and output screens is depicted pictorially using the screen flow chart, a hierarchical structure in which follow-on output screens are shown immediately below the preceding input screen. For example:



In this example SCREENA is an input screen with three possible follow-on output screens: SCREENB, SCREENC, and SCREEND.

The remainder of the SC statement information is specified in the Conditions window.

OPERATION

- XR operator – the desired type 1, 2, or M requests/procedures are specified as Validate, Retrieve, or Update input control procedures, respectively. Only one procedure can be specified for a given type. If additional requests/procedures are required, they must be called from the procedure specified in the Conditions Window.
- EQ, NE, GT, LT, GE, LE operators – these are specified in the Output Conditions section using an IF/END block which consists of an IF statement, followed by one or more OUTPUT or RELEASE statements and terminated by an END statement. For example:

```
IF F.ATTNID EQ 'PF001'
  OUTPUT SCREEN
END
```

- SG operator – specified using the RELEASE statement.
- OS operator – specified using the OUTPUT statement. An output request/procedure can be specified using the CALL operand and an alternate destination can be specified using the DESTINATION operand.
- OC operator – OC specifications are made using the OPTIONS and OUTLIM keywords on the OUTPUT statement.

STATEMENT SEQUENCE – not used.

LOGIC LEVEL – not used. Logical conditions are specified using the IF statement.

CONNECTOR – not used. Logical conditions are specified using the IF statement. Actions designated by the E (ELSE) connector are represented by those OUTPUT or RELEASE statements which are not contained within an IF/END block.

Screen Definition (SD) Statement

SD statement information is specified in the Screen Structure Definition window.

CATALOG ACTION – not used.

GLOSSARY – not used.

HIGHLIGHT DESIGNATOR – the highlight designator is not used in specifying screen layouts. (See [Screen Layout \(SL\) Statement on page B-5](#) for a discussion of highlighting.) However, it is used when applications are converted to VISION:Transact during an export operation. The highlight designator used at that time is specified in the System Parameters window (see [Chapter 2, Getting Started – Installation and Setup](#)).

AUTOMATIC FORMAT – not supported.

OUTPUT CONTROL – you can use the OUTLIM operand of the OUTPUT statement (Conditions window) to emulate this feature.

SCREEN IDENTIFIER – called “BMS/MFS Name” in VISION:Workbench.

Screen Layout (SL) Statement

Text is entered directly onto the screen just as you want it to appear during application execution. Display attributes, such as color and highlighting, are simulated as closely as possible. Those that cannot be simulated by your PC monitor are displayed using reverse video.

LINE NUMBER – not used.

CONTINUATION – not used.

XATTR – display attributes are specified using the Attributes window and subsequently painted onto the screen by you using the cursor control keys.

TEXT – literal text is entered onto the screen just as you want it to appear. Highlighting is achieved using the Attributes window (see XATTR above). Field designators are not entered directly onto the screen. Instead, position the cursor where you want a screen field to appear and open the Contents window. When you close the Contents window, the layout is updated with the appropriate number of field designators. (You can either enter the screen field length explicitly in the Contents window or let VISION:Workbench compute the length based on the input or output fields you specify.) If a screen field occurs multiple times, you enter the first occurrence as described above and use the COPY selection (from the EDIT menu) to create additional occurrences.

Screen Field (S0) and Screen Content (SN) Statements

All the S0/Sn information for a screen field is specified in a single Contents window, including all information relating a screen field to its source or any of its (possibly multiple) destinations. The Contents window for displayable fields is accessed by pressing F8 anywhere in the selected field on the Screen Layout window. Virtual field Contents windows are accessed from the Screen Structure Definition window by pressing F8 on a Virtual Field button field, entering a name for the virtual field, and pressing F8 again to open the Contents window.

VIRTUAL/SAME/AFTER

- *VIRTUAL* – see [Using SCREEN on page 9-16](#) for a description of how to specify virtual fields.
- *SAME* – not used. All specifications for a screen field are made in the same Contents window.
- *AFTER* – not used. If you want screen fields to be displayed after fields in dependent segments, simply position them where you want them to appear on the screen layout.

INPUT/OUTPUT – called “Type” in VISION:Workbench.

INPUT REQUIRED? – you can achieve this specification by checking the screen field for a blank value using the VERIFY statement. For example:

```
VERIFY S.CUSTNO NE ' ' MESSAGE 'CUSTOMER NUMBER REQUIRED'
```

HIGHLIGHTING – specified just as all other display attributes, using the Attributes window (see XATTR in [Screen Layout \(SL\) Statement on page B-5](#)).

CURSOR – use the CURSOR menu selection in the Screen Layout window to specify cursor position. Once you have done so, the display attributes for the cursor position will be reversed to indicate its position. (This does not affect the attributes displayed at application execution time, however.)

ERROR HANDLING – called “Hilite Err” in VISION:Workbench.

NUMERIC – the Num. Lock entry allows you to set the 3270 numeric field attribute for the screen field. VISION:Transact numeric or floating point validation can be achieved using the VERIFY statement in conjunction with the NUMERIC or FLOAT named patterns. For example:

```
VERIFY S.CUSTNO EQ P.NUMERIC MESSAGE 'MUST BE NUMERIC'
```

LOGIC LEVEL – not used. Validation conditions are specified using logical expressions within the VERIFY statement.

CONNECTOR – not used. Validation conditions are specified using logical expressions within the VERIFY statement.

EDIT/VALIDATION OPERATOR

- EQ, NE, LT, GT, LE, GE operators – validation criteria are specified using the VERIFY statement.
- MS operator – a validation failure message can be specified on the VERIFY statement. For example:

```
VERIFY S.CUSTNO NE ' ' MESSAGE 'CUSTOMER NUMBER REQUIRED'
```

- SE operator – fields to be highlighted are specified using the ON statement. For example,

```
ON ERROR HILITE S.FIELD1 S.FIELD2
```

- IF operator – conditions for dynamic actions are specified using the CASE statement. The associated dynamic action is specified using an Action statement immediately following the CASE statement. The default action is specified following an ELSE statement. For example:

```
DO INPUT
CASE F.ATTNID EQ 'PF001'
  UPDATE SEGMENT
CASE F.ATTNID EQ 'PF002'
  DELETE SEGMENT
ELSE
  IGNORE
END
```

- NM operator – specified using the Name entry. Note that a screen field name is *required* in VISION:Workbench, although it is optional in VISION:Transact.
- LN operator – specified using the Length entry.
- ED operator – specified using the Edit Mask entry.
- EN operator – specified using the Num. Edit entry.
- EC/ET operators – for error codes F06, S01, S02, S03, S04, and S05 use the ON statement. For error codes F01, F03, and F04 use the VERIFY statement. For example:

```
ON ERROR NOT_FOUND MESSAGE 'CUSTOMER NOT ON DATA BASE'  
VERIFY S.CUSTNO NE ' ' MESSAGE 'CUSTOMER NUMBER REQUIRED'
```

- DS operator – not supported.
- DF operator – can be emulated by assigning a default value to the blank screen field within a Validate (type 1) Input Control Procedure.

Screen Segment (SS) Statement

These specifications are made using the Screen Structure Definition window.

Temporary Field (TF) Statement

Temporary Fields can be specified on the Temporary Field window (which is accessed from the Request Definition window) or by entering FIELD statements within the Procedure Definition window.

Index

A

ABOVE, 5-16, 9-15
action statement, 10-9
adding new text lines, 4-15
additional validation/actions, 10-8
Advanced Syntax Language, 3-1, 3-4
Application Definition menu bar, 9-10
 EXPORT, 9-10
 PROTOTYPE, 9-10
 RETRIEVE, 9-10
Application Definition window, 6-2, 9-9
application environment, 9-20
application import considerations, 12-10
Application Parameters window, 6-6
Array Comments window, 5-6
Array Definition window, 5-6
array definitions, 5-6
Array List window, 5-6
ASL, 3-1, 3-4, 9-7
ATTRIBUTES, 10-5

B

Button field, 4-7

C

CALL statement, 3-4
CASE statement, 10-9
CENTER, 5-18, 10-4
CHOICES, 4-6, 4-8
CLOSE, 4-6
column headings, 7-5
comment statement, 10-10
Comments window, 9-21
comments window, 4-13
compression off, 2-4
compression on, 2-4
Conditions window, 9-17
 event controlled procedures, 9-13
CONTENTS, 7-9, 7-10
Contents window, 7-9, 10-7
Contents/Action window, 9-17
control procedures, 6-3
 VISION:Builder, 3-2
 VISION:Transact, 9-7, 9-17
conversion utility, 12-4
COPY, 4-11, 4-12, 10-4
CURSOR, 10-6

D

defining

- a new file definition, 5-11
- arrays, 5-6
- column headings, 7-5
- files, 5-2, 5-9
- reports, 7-1
- screen layout, 9-14, 10-1
- screen structure, 9-16
- screens, 9-14
- tables, 5-4
- temporary fields, 9-23
- transactions, 5-7

definition component, 5-1

Definition menu bar, 5-14

- ABOVE, 5-16
- CENTER, 5-18
- COPY, 5-17
- DELETE, 5-17
- EDIT, 5-17
- EXPORT, 5-17
- MOVE, 5-17
- ORPHANS, 5-17
- RELATION, 5-16
- REPLACE, 5-17
- RETRIEVE, 5-17
- SEARCH, 5-17

DELETE, 4-6, 4-16, 4-19, 5-14, 5-17

deleting applications, 4-19

deleting existing text lines, 4-16

deleting objects, 4-19

deleting segments, 5-14

destination fields, 10-8, 10-10

display fields, 9-16

DO INPUT block, 10-8

DO statement, 10-8

documenting applications, 4-13

dynamic actions, 10-10

E

EDIT, 5-17, 10-4

editing menu bar, 4-11

End key, 10-3

END statement, 10-10

entering text, 4-7

environment variable, 2-2

Event controlled procedures

VISION:Builder, 3-3

event controlled procedures

VISION:Builder, 6-3

VISION:Transact, 9-7

EXCHANGE, 10-4

EXPORT, 5-17, 6-14, 9-10

EXPORT facility, 12-10

export input path, 2-4

F

File Comments window, 5-3

File Definition window, 5-3, 5-9, 5-15

file definitions, 5-2, 5-9

File List window, 5-3

Files Used window, 6-5, 9-12

Follow-on screen, 9-19

function keys, 4-5

Alt+F1, 4-6

Alt+F10, 4-6, 4-16, 4-19, 5-14, 9-16

Alt+F3, 4-6, 9-18

Alt+F6, 4-6

Alt+F8, 4-6, 4-10, 10-6

F1, 4-5, 4-9

F2, 4-5

F3, 4-6

F4, 4-6, 4-11

F5, 4-6, 4-18

F6, 4-6, 4-8

F7, 4-6, 4-17

F8, 4-6

F9, 4-6, 4-16

G

GEN, 7-10

H

HELP, 4-5, 4-6

Help information, 4-9

I

import input path, 2-4

importing applications and definitions, 12-4

input control procedures, 9-7

INPUT to, 10-8

INSERT, 4-6, 4-16

inserting text lines, 4-16

installation, 2-1

L

LEVEL, 7-10

library path, 2-4

M

main menu, 1-2

MENU, 4-6

modifying an application, 4-15

MOVE, 4-11, 4-12, 10-4

MPCPROF, 2-2

N

NAMES, 4-5

navigation, 4-1, 5-10

normal execution, 12-11

O

Objects

 VISION:Builder, 3-4

objects

 retrieving, 6-14

 screens, 10-1

 VISION:Transact, 9-8

ON statement, 10-9

OPEN, 4-6

ORPHANS, 5-17

Orphans, 3-2

orphans, 5-14

output conditions, 9-6, 9-18, 9-19, 10-11

OUTPUT from, 10-8

OUTPUT statement, 9-18

P

PAINT, 10-5, 10-6

painted reports, 7-5, 7-8, 7-10

PC display monitor, 2-4

PRINT, 4-6

print initialization, 2-4

print termination, 2-4

print utility, 12-2

printer graphics character support, 2-4

printer output dest, 2-4

printer setup, 2-4

PROCEDURE, 6-14

Procedure Definition window, 6-9, 9-22
procedures vs. requests, 3-1
profile information, 2-2
prompts, 4-10
PROTOTYPE, 9-10, 9-17

R

RELATION, 5-16
RELEASE statement, 9-19
REPLACE, 4-11, 4-12, 10-4
REPORT, 6-14
Report Column Headings window, 7-5
Report Comments window, 7-6
Report Definition menu bar, 7-10
 CONTENTS, 7-10
 GEN, 7-10
 LEVEL, 7-10
 SIMULATE, 7-10
Report Definition window, 6-12, 7-1
Report Details window, 7-4
Report Extended Titles window, 7-8
Report Manager window, 6-13
Report Preface window, 7-3
Report Summaries window, 7-9
Report Title window, 7-7
REQUEST, 6-14
Request Definition window, 6-10, 9-23
requests and procedures, 3-1
RETRIEVE, 5-17, 6-14, 9-10, 9-17
Run Parameters window, 2-7

S

sample reports, 12-11
SAVE, 4-6, 4-18

saving applications, 4-18
SCREEN, 9-16
screen control, 3-6
screen definition, 9-14
Screen Definition window, 9-21
screen flow, 10-2
screen flow menu bar, 9-14
 ABOVE, 9-15
 PROTOTYPE, 9-17
 RETRIEVE, 9-17
 SCREEN, 9-16
Screen Flow window, 9-13
screen layout, 9-14, 9-16, 10-1
screen layout menu bar, 10-4
 ATTRIBUTES, 10-5
 CENTER, 10-4
 COPY, 10-4
 CURSOR, 10-6
 EDIT, 10-4
 EXCHANGE, 10-4
 MOVE, 10-4
 PAINT, 10-5, 10-6
 REPLACE, 10-4
 SEARCH, 10-4
Screen Structure Definition window, 9-16
SEARCH, 4-11, 4-12, 10-4
Segment Definition window, 5-3, 5-15
segment structure chart, 5-9
 navigation, 5-10
setup information, 2-2, 2-4
shift tab key, 10-3
SIMULATE, 7-10, 7-11
simulating reports, 7-11
source fields, 10-8
specifying actions, 10-10
standard reporting, 7-5
SUBFILE, 6-14
Subfile Definition window, 6-7

summary only report, 7-9
SWITCH, 4-6, 4-7, 4-10, 10-6
syntax validation, 12-11
system parameters, 2-7
system requirements, 2-1

T

tab key, 10-3
Table Comments window, 5-5
Table Definition window, 5-5
table definitions, 5-4
Table List window, 5-4
Target Mainframe Support, 2-5
temporary fields, 6-11, 9-23
Transaction Comments window, 5-8
Transaction Definition window, 5-8, 5-20
transaction definitions, 5-7, 5-18
 import considerations, 12-9
Transaction Group window, 5-7
Transaction Groups Used window, 6-6
Transaction List window, 5-7
transactions, 3-5

U

utilities, 12-1

V

VALIDATE, 4-6, 4-17
validating applications, 4-17
validation statements, 9-5
VERIFY statement, 10-9
virtual fields, 9-16
VISION:Builder Application List Window, 6-1

VISION:Builder applications, 6-1
 Application Definition window, 6-2
 Application Parameters window, 6-6
 Column Headings window, 7-5
 Comments window, 7-6
 Contents window, 7-9
 control procedures, 6-3
 creating reports, 7-1
 event controlled procedures, 6-3
 Files Used window, 6-5
 Procedure Definition window, 6-9
 relationship to VISION:Workbench, A-1
 Report Definition window, 6-12, 7-2
 Report Details window, 7-4
 Report Extended Titles window, 7-8
 Report Preface window, 7-3
 Report Summaries window, 7-9
 Report Title window, 7-7
 Request Definition window, 6-10
 simulating reports, 7-11
 Subfile Definition window, 6-7
 Transaction Groups Used window, 6-6
VISION:Builder processing control procedures, 3-2
VISION:Builder processing transactions, 3-5
VISION:Transact applications, 9-6
 Comments window, 9-21
 Contents window, 10-7
 destination fields, 10-8, 10-10
 dynamic actions, 10-10
 environment, 9-20
 Files Used windows, 9-12
 Follow-on screen, 9-19
 output conditions, 9-19
 Procedure Definition window, 9-22
 relationship to VISION:Workbench, B-1
 Request Definition window, 9-23
 Screen Definition window, 9-21
 Screen Flow window, 9-13
 screen layout, 10-1
 source fields, 10-8
 specifying actions, 10-10
 validation/actions, 10-8

VISION:Transact input control procedures, 9-7,
9-17

VISION:Workbench for DOS

- EXPORT facility, 12-10

- features, 1-1

- main menu, 1-2

- print utility, 12-2

- setup information, 2-4

- structure, 1-2

- system requirements, 2-1

- VISION:Transact applications, 9-6

- walk-through (VISION:Builder), 8-1

- walk-through (VISION:Transact), 11-1

W

walk-through

- VISION:Builder, 8-1

- VISION:Transact, 11-1

Z

ZOOM, 4-6, 9-16, 9-18