Advantage™ VISION:Builder®

Advantage™ VISION:Two™

Reference Summary

14.0

Computer Associates™

BUSPC140.PDF/D92-007-014

Contents

Chapter 1: Overview of VISION:Builder Statements

Chapter 2: Fixed Format Statement Listing

Chapter 3: Flags

Chapter 4: VISION:Builder Technical Notes

Chapter 5: Graphic Fixed Format Statement Listing

Index

Chapter

1 Overview of VISION:Builder Statements

How the Guide is Organized

The *Specifications Guide* contains all valid specifications for VISION:Builder® statements.

- Chapter 1- Describes guide contents, M4PARAMS and MARKLIB, file summary, VISION:Builder runs and statements, general rules for VISION:Builder statements, and how to invoke ASL (Advanced Syntax Language).
- Chapter 2, - Alphabetically lists all non-graphic statement types. To make reference easy, the name of the statement code is printed on the header of each page.
- Chapter 3, - Describes flags used in processing and reporting statements.
- Chapter 4, - Reviews technical specifications.
- Chapter 5, - Alphabetically lists all graphic statement types. To make reference easy, the name of the statement code is printed on the header of each page.

Model and Options

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

- The 4000 model series represents the VISION:Builder software system that allows users to develop fully functional applications that include the transaction processing and master file updating facilities along with the entire set of information retrieval, selection, and manipulation functions used in conjunction with the multitude of reporting and data extraction capabilities.

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

The term VISION:Builder is used to represent both the 4000 model series and the VISION:Two 2000 model series. The model number appears on the banner page that is output at the start of all source listing displays. Throughout this guide, the symbol 4 is used to designate that the feature or function being described only applies to the VISION:Builder 4000 model series.

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

- The IMS option provides support for processing information in IBM® IMS Databases using the standard DL/I processing facilities. Throughout this guide, the symbol I is used to designate that the specification, feature, or function being described only applies to the IMS option.
- The DB2 option provides support for processing information in IBM Database2 Relational Tables using the standard SQL processing facilities. Throughout this guide, the symbol D is used to designate that the specification, feature, or function being described only applies to the DB2 option.
- The GDBI option provides a facility for interfacing user code with the standard mechanisms of VISION:Builder to perform I/O operations using any database processing facilities. Throughout this guide, the symbol G is used to designate that the specification, feature, or function being described only applies to the GDBI option.

The options selected appear on the banner page that is output at the start of all source listing displays. In the documentation, there are margin notes indicating which of the VISION:Builder specifications apply to a particular option.

M4PARAMS and MARKLIBP

VISION:Builder provides standard default values for the parameters that are usually determined by the operating environment in each installation. These parameters establish standard page height and width, print line width, and system delimiter. With VISION:Builder, you can use the default values or select your own.

Two special program modules called M4PARAMS and MARKLIBP define each parameter and assign the standard default.

M4PARAMS is supplied with VISION:Builder and MARKLIBP is supplied with COMLIB. The following table lists all the parameters and their default values. The table also includes space for you to fill in the override values that your installation selects. The sign-on page lists the M4PARAMS and MARKLIBP parameters installed in your system.

M4PARAMS Parameters	As Delivered	Install Override
User ID	32 characters of text in signon	
System Delimiter	# (the underscore (_) and tilde (~) are illegal override characters).	
M4LIST Maximum Height of Page	66 (11 inch page at 6 lines/inch)	
M4LIST Maximum Width of Page	132 columns	
M4LIST Default Width of Page	0 – The default page width must be less than or equal to the standard page width. If the default width of page is zero, it will be assumed to be equal to the M4LIST page width.	
Automatic Grand Summaries	No autogrand summary page is printed.	
Report Column Heading Character	- (dash or hyphen)	
S-Type Subtitle Control	No repeated subtitles on page overflow.	

M4PARAMS Parameters	As Delivered	Install Override
Special Output Characters		
Invalid	* (field is invalid)	
Missing	- (field does not exist)	
Uneditable	+ (field cannot be edited)	
Percent Character	%	
Summary Labels		
Total	TOTAL	
Cumulative	CUM.	
Count	COUNT	
Maximum	MAX.	
Minimum	MIN.	
Average	AVG.	
Ratio	RATIO	
Percent	PCT.	
Grand	GRAND	
Page Label	PAGE	
Source Statement Listing Delimiters		
Left Separator	((left parenthesis)	
Right Separator	) (right parenthesis)	
Single Separator	, (comma)	
Vertical Spacing	Single Spaced	
Message Control		
Printer Messages	On	
Console Messages	Off	
M4REPO Block Size	4096 bytes (1024 bytes for VSE)	

M4PARAMS Parameters	As Delivered	Install Override
Number of I/O Buffer Files		
Input		
■ OS/390® & VSE	2	
■ CMS	1	
Output	1	
One-Step Report Storage	8K bytes for OS/390 24K bytes for CMS 32K bytes for VSE	
One-Step Sort Storage	512K bytes 40K bytes for VSE	
Digit Selection Character	9	
Zero Suppress Character	Z	
Currency Symbol	$	
Plus Symbol	+	
Minus Symbol	-	
Check Protection Character	*	
Decimal Point	.	
Grouping Character	,	
Graph Primary Plot Character	X	
Graph Secondary Plot Character	*	
Graph Fit Plot Character	.	
Graph Horizontal Axis Character	_ (underscore)	
Graph Horizontal Hash Character	|	
Graph Vertical Axis Character	|	
Graph Vertical Hash Character	- (dash)	

M4PARAMS Parameters	As Delivered	Install Override
Unit Conversion Multipliers and Delimiters		
Minutes/Hour	60	
Seconds/Minute	60	
HH:MM:SS Separator	: (colon)	
Months		
January	JAN	
February	FEB	
March	MAR	
April	APR	
May	MAY	
June	JUN	
July	JUL	
August	AUG	
September	SEP	
October	OCT	
November	NOV	
December	DEC	
DATE Flag Format	MMM DD, YYYY	
TODAY Flag Format	MMDDYY	
TODAY Flag Delimiter	/ (MM/DD/YY)	
ISDATE Flag Delimiter	- (YYYY-MM-DD)	
JULIAN Flag Delimiter	. (YY.DDD)	
SORT Program		
OS/390	5740-SM1	
CMS	5734-SM1	
VSE	5746-SM1	
MAXGETMN Working Storage Size	1024K bytes	

M4PARAMS Parameters	As Delivered	Install Override
Minimum Storage Release to System	12K bytes	
Alternate M4LIST Maximum Width of Page (24 minimum)	132 columns	
Alternate Default Width of Page	0 – Causes default to M4LIST1	
Suppress No Data Selected Report	N No data selected reports will be printed.	
DECMSOPT	Y	
PROMSOPT	Y	
RPTMSOPT	Y	
Default File Processing AMODE(31)	Y	
PAL Max Lines of Trace	1,024	
High Level ISAM Index Control - OS/390	0 No indices in storage	
CONDCOD1	0 (normal termination)	
CONDCOD2	4 (error termination)	
CONDCOD3	8 (no sorting - RC specification)	
CONDCOD4	16 (no sorting - invalid requests)	
Number of Tracks for ISAM Cylinder Overflow - VSE	2	
M4LIST Unit Assignment - VSE	Attempt SYS026 Assign	
M4INPUT Unit Assignment - VSE	Attempt SYS025 Assign	
M4REPO Volume Count VSE	0 Volumes	
M4LIST Tape Labels - VSE	Unlabeled Tape	

MARKLIBP Parameters	As Delivered	Install Override
COMLIB Directory		
Blocking Factor (BDAM COMLIB only)	0 – A suitable factor will be calculated based upon device type and tracks allocated.	
Device Reserve flag	0	
Item Tracking flag	0 – No optional item tracking features enabled	
Minimum Compress Size	507	
Compression flag	0	

File Summary

The File summary page provides OS/390 users data set information for the application. The output includes the job name and procedure names/step names, as well as the ddnames and data set/DBD names for certain VISION:Builder files, including M4OLD, M4NEW, M4TRAN, M4CORDn, M4SUBFn, and M4LIBn. The File summary page prints directly following the application run control group. A sample File summary page is shown in Figure 1-1.

```
                F I L E    S U M M A R Y

JOB NAME: ISPEJM          STEPNAME:

FILE NAME: M4OLD          DSNAME: PROD.CUSTOMER.DATA
FILE NAME: M4CORD1        DSNAME: PROD.ITEM.DATA
FILE NAME: M4CORD2        DSNAME: DEVELOP.FINANCE.DATA
FILE NAME: M4CORD3        DSNAME: DEVEL.ORDER.FILE
FILE NAME: M4SUBF1        DSNAME: TEST.SUBFILE.ONE
FILE NAME: M4SUBF2        DSNAME: TEST.SUBFILE.TWO
FILE NAME: M4SUBF3        DSNAME: TEST.SUBFILE.THREE
FILE NAME: M4LIB          DSNAME: PROD.COMMON.LIBRARY
```

Figure 1-1 Sample File Summary Page

VISION:Builder Runs and Statement Order

Figure 1-2 should help you assemble and organize your VISION:Builder runs.

- Note that an RC statement precedes all statements in any run except a MARKUTIL run (which accepts only UC statements).
- The AA (comment) statement can be used anywhere in the run after the RC statement.

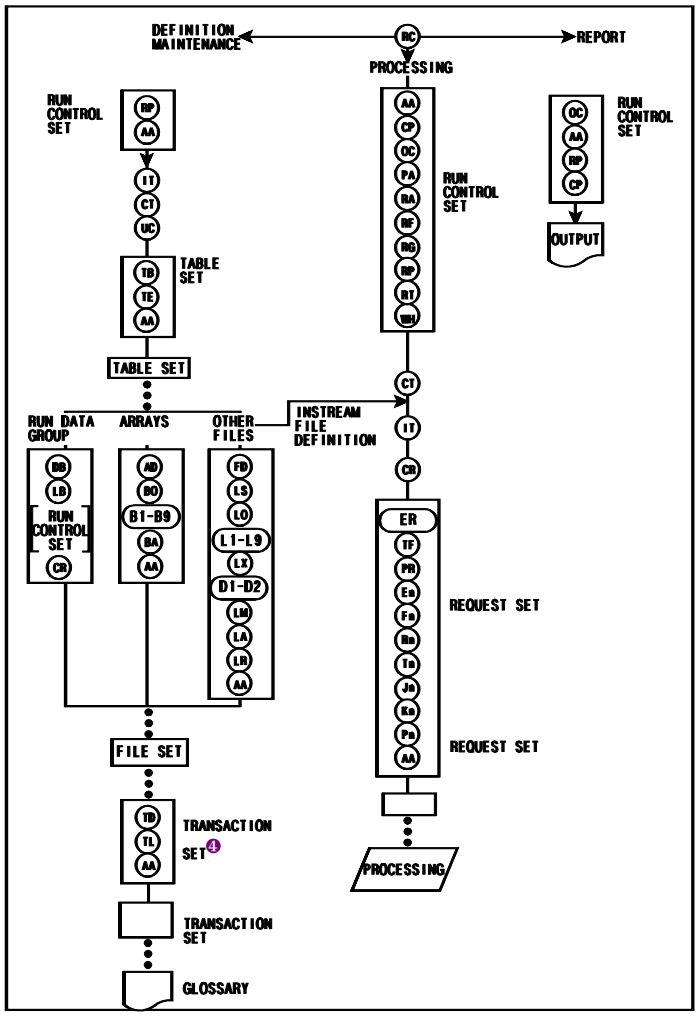

Figure 1-2 VISION:Builder Statement Types

General Rules for VISION:Builder Statements

For shorthand references, the following rules are used in Chapter 2, *Fixed Format Statement Listing*.

Numeric Entries: Right-aligned.

Alphabetic Entries:Left-aligned.

Shorthand Reference	Meaning
Rule A	File names, field names, master files (old and new 4), alternate M4LIST names, request names, run names, transaction groups, 4 and table names use a unique name consisting of 1 to 8 left-aligned characters. First character must be alphabetic; others can be any combination, including special characters or blanks. Do not use system delimiters or override delimiter characters.
Rule B	Blank or numeric.
Rule C	Blank, N, or Y are legal entries for "?" specifications.
Rule D	Numeric.
Rule E	Use left-aligned alphanumeric characters. Embedded blanks are valid.

Shorthand Reference	Meaning	
Rule F	Qualifiers.	
	Blank or N	New master file work area
	O or 0	Old master file work area
	1 to 9	Coordinated files 1 to 9
	T	Temporary field
	F	Flag
	X	Transaction file[4]
	V	Linkage section field
	W	Working storage field
	A, B, E, G, H, J, K, M, Q, 1 to 9	Array
Note: For Rules G and H, OS/390 users must use one of the codes, but the function is achieved by JCL.		
Rule G	Data file labels (audit file,[4] coordinated files, master file, rejected transaction file,[4] source statement file).	
	Blank	No file
	S	Standard labels
	U	Unlabeled
	X	Unlabeled, no rewind
	V	Unlabeled, multi-volume tape
	N	Non-standard

Shorthand Reference	Meaning	
Rule H	Report file labels.	
	Blank	No report (In/Out)
	S	Standard labels
	U	Unlabeled
	V	Unlabeled, multi-volume tape
Rule I	Statement sequence.	
	Blank, or enter up to 3 alpha, numeric, or alphanumeric characters to specify either the order in which tasks are to be processed or the line sequence wanted for lines of title. Allow gaps in numbering so insertions can be made.	
	Sequence numbers are treated as character strings; consequently b/20, 020 and 20/b are not equal.	
Rule J	Abbreviated format.	
	Blank or N	Full display of all data
	R	Suppress repeated selections of same data field
Rule K	Special handling of input records.	
	Blank	Fields on Rn to be output to report or subfile
	N	New master file work area output to subfile
	O or 0	Old master file records output to subfile
	1 to 9	Coordinated file or array records output to subfile
	X	Transaction file records output to subfile ❹
	A, B, E, G, H, J, K, M, Q	Array records output to a subfile

<table>
<tr><th>Shorthand Reference</th><th>Meaning</th></tr>
<tr><td>Rule L</td><td>Time format
<table>
<tr><th>Field Type</th><th>Time Representation</th></tr>
<tr><td>L</td><td>Signed packed decimal format: hh-hmmssnn-n
hh-h = hours
mm = minutes
ss = seconds
nn-n = decimal of seconds (up to 9 digits)</td></tr>
<tr><td>S</td><td>Signed packed decimal format: mm-mssnn-n
mm-m = minutes
ss = seconds
nn-n = decimal of seconds (up to 9 digits)</td></tr>
<tr><td>U</td><td>Zoned decimal format: hh-hmmssnn-n
hh-h = hours
mm = minutes
ss = seconds
nn-n = decimal of seconds</td></tr>
<tr><td></td><td>Field type U is an external format and cannot be stored internally.</td></tr>
<tr><td></td><td>Minimum and maximum transaction editing is not performed.</td></tr>
</table></td></tr>
<tr><td>Rule M</td><td>Type codes
<table>
<tr><td>Blank</td><td>All item types</td></tr>
<tr><td>AD</td><td>Array definition</td></tr>
<tr><td>DB</td><td>Logical run group</td></tr>
<tr><td>FD</td><td>File definition</td></tr>
<tr><td>RQ</td><td>Request</td></tr>
<tr><td>RG</td><td>Request group</td></tr>
<tr><td>TB</td><td>Table definition</td></tr>
<tr><td>TD</td><td>Transaction group definition[4]</td></tr>
</table></td></tr>
</table>

Defining ASL Procedures

ASL is the free-form language used to build procedures consisting of logical, arithmetic, branching, and text processing statements. ASL procedures can be used in place of traditional VISION:Builder requests. ASL can be used in VISION:Workbench™ or entered instream with traditional VISION:Builder requests. ASL procedures and traditional requests must be maintained as separate units.

Do not mix ASL procedure statements and traditional statements in association with the same ER for record selection and processing.

ASL statements are entered after an ER statement. The next line must contain two system delimiters, starting in column one, immediately followed by PROC. The line after the ASL statements must contain two system delimiters, starting in column one, immediately followed by PEND.

The following is an example of in-stream ASL statements:

```
reqname ER
##PROC
   .
   .
ASL statements
   .
   .
##PEND
```

ASL is fully documented in the *ASL Reference Guide*.

Reviewing Workbench Output Using an ISPF Editor

If you review statements exported from VISION:Workbench for DOS or VISION:Workbench for ISPF using an editor displaying the first 72 columns of the exported statement, you may encounter statements and codes that are not documented in this guide. These statements and codes are for internal use only and will not adversely affect your application in any way. Do not remove or modify these statements and/or codes. We strongly recommend that you use the appropriate Workbench component to make any changes to these statements.

Chapter

2 Fixed Format Statement Listing

This chapter contains a description of each VISION:Builder statement. The VISION:Builder statements are as follows:

- RG Statement - Run Control Group¼
- RP Statement - Run Control Group
- TB Statement - Table Definition
- TE Statement - Table Definition
- TL Statement - Transaction Definition¼
- UC Statement - Library Maintenance
- Wn Statement - Output Specification
- Rn Statement - Output Specification
- RT Statement - Run Control Group
- TD Statement - Transaction Definition¼
- TF Statement - Temporary Field Statement
- Tn Statement - Title/Preface/Formatted Reporting
- WH Statement - Run Control Group

AA Statement - Title/Preface/Formatted Reporting

The comment statement (AA statement) allows you to document your application. Comment statements are not processed or cataloged with definitions but are cataloged with requests. The data prints exactly as entered on the statement. You can comment any item by following a statement with one or more AA statements.

Position	Entry
01-08	Request name
09-10	Statement type
11-13	Statement sequence
14-72	Text

Request name (01-08)

Required entry

Identifies the item to which this statement belongs. See general Rule A.

Enter the same name for all AA statements in the item.

Statement type (09-10)

Required entry

Identifies a comment statement. Enter AA.

Statement sequence (11-13)

Not used on the AA statement. Comments are accepted in order of input.

Text (14-72)

Enter comments exactly as they are to appear.

AD Statement - Array Definition

The array definition (AD) statement names and defines physical (record format) and logical (number of rows and columns) array characteristics.

Position	Entry
01-08	Array name
09-10	Statement type
11-18	File identification
19	Delete?
20	Glossary
21	Record format
22-25	Record size
26-28	Records per block
29-33	Buffer size
34-39	Number of rows
40-45	Number of columns

Array name (01-08)

Required entry

Name of the array. See general Rule A. Enter a unique 8-character name for each array definition.

Statement type (09-10)

Required entry

Identifies this as the header statement for an array definition. The header statement describes the characteristics of the array to VISION:Builder. Enter AD.

File identification (11-18)

Not used.

Delete? (19)

Deletes an existing array definition.

Entry	Result
Blank or N	Definition is added to the library.
Y	Definition is deleted from the library.
R	Definition is replaced if it is already cataloged; otherwise, it is added to the library.

Glossary (20)

Note: When requesting a glossary, any entry in the AD statement other than the array name, statement type, and glossary type code results in creating or deleting the array definition.

Selects either of two glossary listings or none at all. See Figure 2-1 and Figure 2-2 at the end of the section on the AD statement. The glossary listing includes the file characteristic information.

Entry	Glossary Type
Blank or A	Alphabetically by field name in each data cell.
B	Sequentially by field location in each data cell.
N	No glossary.

Record format (21)

Required entry

Note: If any file characteristics are specified (for example, records per block), the record format must also be specified. However, if record format is specified, the other file characteristics can be left blank and VISION:Builder will compute them.

Identifies the physical record format and the access method to the operating system. Enter one of the following:

Entry	Record Format
Blank or F	Fixed length.
V	Variable.
K	Key sequenced VSAM.
E	Entry sequenced VSAM.
I	Indexed sequential (fixed blocked).

Record size (22-25)

Defines the size of the array to VISION:Builder. The array size is computed by multiplying the size of one data cell (the last defined field position determines the size of the data cell) times the number of rows, times the number of columns, plus the key field size.

Note: When specifying an external array with F, I, or unblocked V formats, the block size (the product of record size and records per block) must be at least 1.

Array definitions for external files must conform to the system's record size requirement. If the array is too small or too large to be used with an external file, record characteristics are not output in the glossary.

Entry	Result
Blank	VISION:Builder computes array size.
1–9999	Exact size of the array.

Records per block (26-28)

Number of records per block.

Format	Records Per Block
F, I	Blank or 1–999.
V	Blank.
K, E	Blank.

Note: If record size is specified, records per block must also be specified. If both fields are left blank, the array size is computed and the blocking factor defaults to 1.

Buffer size (29-33)

Specifies the amount of main storage required for the buffer, the block size, or the logical record.

- Enter all values right-aligned.
- Leading zeros are not required.
- 1–32,760 or nnnnK, where n = multiples of 1024 bytes.

Note: Array definitions for external files must conform to the system's record size requirement. If the array is too small or too large for an external file, record characteristics are not listed in the glossary.

Record Format	Buffer Size Entry
F, I	Enter the block size.
V	Maximum buffer size (excluding RDW and BDW). If variable block spanned with a record size greater than 32,760, code maximum record size.
K, E	Maximum record size (according to the VSAM define cluster specification if VSAM).

Number of rows (34-39)

Required entry

Number of rows in the array. Enter a number from 1–999999.

Number of columns (40-45)

Required entry

Number of columns in the array. Enter a number from 1–999999.

```
JAN 15, 1995  12.33.05                                                                                                    Page 1

                                        DETAILED GLOSSARY BY NAME FOR
                                        ARRAY DEFINITION - ARRAYMFG

DATE CATALOGED -                                                          DATE LAST UPDATED -
CATALOGED BY -                                                            LAST UPDATED BY -
EXPIRATION DATE -                                                         DATE LAST USED -

NUMBER OF ROWS = 2                                                        RECORD FORMAT = FIXED BLOCKED  (DEFAULT)
NUMBER OF COLUMNS = 2                                                     RECORD SIZE = 91               (COMPUTED)
ARRAY SIZE = 91                                                           RECORDS PER BLOCK = 1          (DEFAULT)

-------------------------------------------------------------------------------------------------------------------------
FIELD    FIELD    FIELD   FIELD   FIELD    DEC    ARRAY   EDIT CODES    EDIT     OUTPUT  LINE      REPORT FIELD
NAME     LOCATION LENGTH  TYPE    RNDING   PLACES KEY     ( ) ( ) ( )   LENGTH   WIDTH   NO   ***    HEADING   ***
-------------------------------------------------------------------------------------------------------------------------
ARRAYKEY 1        1       C                        1                     1        1
-------------------------------------------------------------------------------------------------------------------------

DATACELL 1        15      C                                              15       15

DIVISION 1        1       C                                              1        4
                                                                                    1    ***    DIV       ***

PRODUCT  2        1       C                                              1        4
                                                                                    1    ***    PROD      ***
                                                                                    2    ***    NO        ***

QUANTITY 3        3       P                                              7        7
                                                                                    1    ***    UNITS     ***
                                                                                    2    ***    SOLD      ***

TOTL-SLS 6        5       P               2              ($) ( ) ( )    13       13
                                                                                    1    *** TOTAL SALES ***
```

Figure 2-1 Array Definition Glossary - Type A or Blank

JAN 15, 1995 12.33.05 Page 1

DETAILED GLOSSARY BY LOCATION FOR
ARRAY DEFINITION - ARRAYMFG

DATE CATALOGED -
CATALOGED BY -
EXPIRATION DATE -

DATE LAST UPDATED -
LAST UPDATED BY -
DATE LAST USED -

NUMBER OF ROWS = 2
NUMBER OF COLUMNS = 2
ARRAY SIZE = 91

RECORD FORMAT = FIXED BLOCKED (DEFAULT)
RECORD SIZE = 91 (COMPUTED)
RECORDS PER BLOCK = 1 (DEFAULT)

FIELD NAME	FIELD LOCATION	FIELD LENGTH	FIELD TYPE	FIELD RNDING	DEC PLACES	ARRAY KEY	EDIT CODES ( ) ( ) ( )	EDIT LENGTH	OUTPUT WIDTH	LINE NO	***	REPORT FIELD HEADING	***
ARRAYKEY	1	1	C			1		1	1				
DIVISION	1	1	C					1	4	1	***	DIV	***
DATACELL	1	15	C					15	15				
PRODUCT	2	1	C					1	4	1	***	PROD	***
										2	***	NO	***
QUANTITY	3	3	P					7	7	1	***	UNITS	***
										2	***	SOLD	***
TOTL-SLS	6	5	P		2		($) ( ) ( )	13	13	1	***	TOTAL SALES	***

Figure 2-2 Array Definition Glossary - Type B

BA Statement - Array Definition

The BA statement specifies the array name, its type, and the input argument for automatic table lookup. BA statements are used in conjunction with B0 and B1–B9 statements.

Position	Entry
01-08	Array name
09-10	Statement type
11-18	Field Name
30	Field search type
44-51	Table name
52-59	Input argument name

Array name (01-08)

Required entry

Name assigned to the array definition. Enter the 8-character array name on the AD statement.

Statement type (09-10)

Required entry

Identifies this as an automatic array table lookup statement. BA statements are used in conjunction with B0 and B1–B9 statements. Enter BA.

Field name (11-18)

Required entry

Gives a name to the automatic table lookup result field in an array. This field name must appear in all B0, BA, and B1–B9 statements defining the result field. See general Rule A.

Field search type (30)

Indicates the type of table lookup being performed against an argument.

Code	Meaning
Blank or E	Return results of equal argument value only.
N	Return results of equal or nearest argument value.
S	Return results of equal or smaller argument value.
B	Return results of equal or bigger argument value.
I	Return results of linear interpolated argument value.

Note: On a TB statement, table lookup type must be binary for any entry besides E or blank and the argument and result data types must be numeric.

Table name (44-51)

Required entry

Identifies the table for automatic table lookup. See general Rule A. This table name must exist in the same library in which the array is cataloged.

Note: Table lookup result field characteristics are determined by the table definition TB statement. A field in result entry assumes the characteristics of the result. If table result field characteristics are modified, the array definition must also be modified by deleting and recataloging it.

Input argument name (52-59)

Required entry

The name of the field in this array definition that is used as an input argument to the tables. See general Rule A.

This field name must already exist in the array definition.

The input argument field contains the value that is searched for during table lookup.

Bn Statement - Array Definition

The Bn statement defines the column heading text of a field specified by the B0 statement.

Position	Entry
01-08	Array name
09-10	Statement type
11-18	Field name
44-59	Array report heading text

Array name (01-08)

Required entry

Name assigned to the array definition. Enter the 8-character array name on the AD statement.

Statement type (09-10)

Required entry

Identifies this as a line of column heading text for the B0 statement with the same field name. In automatic table lookup, the Bn statements define the array report heading text for a lookup result field. Enter a Bn (where n is a number from 1–9).

Note: Blank lines in the report heading can be created by skipping line numbers (n) where the blank lines are to appear. If a line number is duplicated, the first line number is accepted; subsequent lines are rejected as errors and a message prints accordingly.

Field name (11-18)

Required entry

Name of the field whose report headings are being defined. In an array definition, each field name must be unique and reentered for each line of report heading text. Enter the 8-character field name, exactly as in the B0 statement for this field.

Array report heading text (44-59)

Defines the heading printed over a field on a report. Enter up to 16 characters for each line of field heading.

The text itself can consist of any combination of characters except the system delimiter (the default is #), which indicates the end of a line of heading text. Lines are centered based on the placement of the delimiter, which must be used if trailing blanks are required for the headings.

B0 Statement - Array Definition

The B0 statement defines the attributes of a field in an array (for example, size, data type, and output editing). A B0 statement is required for each array field defined.

Position	Entry
01-08	Array name
09-10	Statement type
11-18	Field name
23-26	Field location
27-29	Field length
30	Field/search type
31	Array key
32	Field rounding
33	Decimal places
39-43	Output edit
39	Floating
40	Filling
41	Trailing
42-43	Length

Array name (01-08)

Required entry

Enter the same 8-character name entered on the AD statement.

Statement type (09-10)

Required entry

B0 identifies this as a statement that defines the attributes of an array field. Enter B0.

Field name (11-18)

Required entry

The name of the field defined. Each field name must be a unique 8-character name. Any field or part of a field can be overdefined with different names and attributes. See general Rule A.

Field location (23-26)

Required entry

Indicates the position of the field in a data cell. Enter a number, right-aligned, from 1–9999, where 1 is the first byte of the data cell. Leave blank for table lookup result field.

Field length (27-29)

Required entry

Field Type	Range
Blank or C	1–255 bytes (characters).
Z	1–15 bytes.
P	1–8 bytes (15 digits plus sign); enter number of bytes, not digits.
E	4 bytes only.
F	1–4 bytes with ranges as follows: Field Length — Range 1 byte — -128 to 127 2 bytes — -32,768 to 32,767 3 bytes — -8,388,608 to 8,388,607 4 bytes — -2,147,483,648 to 2,147,483,647
D	4 bytes only (interpreted as a Lilian date when displayed).

Field/search type (30)

Required entry

Indicates the type of data in the field.

Code	Definition
Blank or C	Character string.
Z	Number in zoned format.
P	Number in packed format.
F	Fixed point binary number (negative values are stored in 2's complement notation).
E	Floating point number.
R	Table lookup result field.
L, S	Time processing.

Array key (31)

Identifies an array key field.

Entry	Meaning
Blank	Fields that are not array keys.
1	Field that is an array key.

Note: An array key must be defined if it is input from or output to an external file that requires a record key.

The array key field appears in the body of the glossary, separated from the data cell fields by dashes. It is not part of the data cell, but occurs once immediately preceding the data cell. Only one key field is allowed; it must be the first field in the array definition and must have a field location value of one. The B0 and Bn statements for the key field must precede all other B0 and Bn statements in the array definition.

Field rounding (32)

Used only for results of arithmetic, replacement, and table lookup. If this field is used as operand C on a PR statement, the contents are rounded, not truncated.

Entry	Rounding?
Blank or N	No
Y	Yes

Decimal places (33)

Indicates the number of decimal places in a numeric field (types Z, P, or F). Leave blank for nonnumeric fields.

Entry	Decimal Places	Field Type
Blank	0 (assumed)	Z, P, F
0–9	0–9	Z, P, F

Note: The number of decimal places can never exceed the maximum number of digits of a field.

For type F fields, the maximum number of decimal places is a function of the field length, as shown in the following table.

Length (bytes)	Maximum
1	3
2	5
3	7
4	9

Output edit (39-43)

The output edit entries (floating, filling, trailing) are only for numeric fields (types P, Z, and F). They specify special editing when the data in the field is printed in a report.

Commas print in all fields when preceded by a significant digit. To suppress printing of commas, use the edit suppress character or the override "picture" edit on the Rn statement [53-67].

Make no entry in the output edit columns for floating point numbers. They always print in a standard notation. The output edit length must exceed 6 to include the 2 signs, decimal point, E, and 2 exponent digits. Standard notation for floating point numbers, where Xs represent the fraction and Ys the exponent, is:

```
±.XXXXXXXE±YY
```

Floating/edit suppress

An entry other than Z in this column causes the value of the entry to "float" and print to the immediate left of the first significant digit in the report.

- If this column is blank, a leading blank is printed if the field is positive; a minus is printed if the field is negative.
- If this column contains a Z, all commas, leading zeroes to the left of the decimal point, and the decimal point, if specified, are suppressed.

Floating (39)

Edit Code	Result
Blank	When no output edit codes are specified, a leading blank is printed if the value of the field is positive; a leading minus sign is printed if the value of the field is negative; commas and decimal points are printed if decimal places are specified. A zero value in a field where a decimal place has been specified is printed as a decimal point followed by as many zeroes as there are decimal places.
$	A floating dollar sign is printed before the first value when a control break occurs and when summaries are taken.
+	A leading plus sign (+) is printed if the value of the field is positive, a leading minus sign if negative.
-	A leading minus sign (-) is printed if the value of the field is negative.
Z	The printing of commas, decimal points, and leading zeroes to the left of the decimal point is suppressed.
Any other character	A floating lead character is printed at control breaks and on summaries.

Note: Minus signs print, but no space is allocated for them.

Filling (40)

This entry prints in every position, left to right, until the first nonzero is encountered.

Code	Results
Any character	Replaces leading blanks in zero positions.

Trailing (41)

This entry prints a trailing character.

Code	Results
+	A trailing plus sign (+) is printed if the value of the field is positive, a trailing minus sign if negative.
-	A trailing minus sign (-) is printed if the value of the field is negative.

Code	Results
)	Negative field values are enclosed in parentheses. ■ If no filling character is specified, the left parenthesis is printed before the first significant digit or decimal point, whichever is first. ■ If a floating character and this character are specified, both can print. The floating character prints inside the parentheses ($43.50). Only a single floating sign is permissible with the trailing ")".
C	Prints a trailing "CR" for a negative value. Blanks follow a positive value.
D	Prints a trailing "DB" for a negative value. Blanks follow a positive value.
Any other character	Prints as a trailing character for negative values only.

Length (42-43)

Specifies the number of characters output for a field.

Field Type	Entry	Resulting Output
Fixed length	Blank	VISION:Builder computes the length.
	Any number equal to the number of positions necessary to print the field.	You compute the length.

Field Type	Number of Bytes	Number of Digits
C	L	L
Z	L	L
P	L	2L - 1
E	4	7
F	1	3
	2	5
	3	7
	4	9

Note: In this table, L stands for the length of the field.

CP Statement - Run Control Group

The checkpoint statement (CP statement) provides checkpoints on VISION:Builder runs. In case of computer malfunction or operator error, the runs can be restarted without excessive loss of processing time.

The checkpoint statements are entered after the RC statement and before the first request in the run.

Note: CMS performs the DB2 Commit Only processing for this statement.

Position	Entry
01-08	Run name
09-10	Statement type
11	Checkpoint file
12	Alternating checkpoint file
13	Sort checkpoint option?
14-16	Time interval
17-30	Record interval on file
17-22	Count
23-30	Control file
31	Operator control?
32-39	End of volume on file(s)
40-46	IMS checkpoint ID prefix
47	DB2 commit only?

Run name (01-08)

Required entry

Identifies this statement as belonging to a particular run group. Enter the name as entered on the RC statement.

Statement type (09-10)

Required entry

Identifies a checkpoint specification statement. Enter CP.

Checkpoint file (11)

Specifies labeling conventions for the file on which checkpoints are written.

Entry	Result
Blank or S	Standard labels.
U	Unlabeled.
X	Unlabeled, but no rewind on OPEN or CLOSE.

Note: This file can be assigned to a tape drive or a disk drive.

When using VSE, the VISION:Builder symbolic unit for this file is SYS027; the DTF name is M4CHK1.

When using OS/390 or IMS, this specification is checked for validation purposes only. The VISION:Builder ddname for this file is M4CHK1. The data set can be sequential or partitioned.

Alternating checkpoint file (12)

Specifies labeling conventions for the file on which checkpoints are written, alternating with the checkpoint file.

When using IMS, ignore this specification.

OS/390:

Entry	Result
Blank	Alternating checkpoints are not written.
S, U, or X	An alternating checkpoint file exists in this run. Each entry is valid and is checked for validation purposes only.

Note: The VISION:Builder ddname for this file is M4CHK2. The data set can be sequential or partitioned and is opened and closed at each checkpoint. When writing checkpoints on alternating files, only one checkpoint is retained on each file unless DISP=MOD is specified in the OS/390 DD statement.

VSE:

Entry	Result
Blank	Alternating checkpoints are not written (only one checkpoint file is used).
S	Standard labels.
U	Unlabeled.
X	Unlabeled, but no rewind on OPEN or CLOSE.

Note: This file can be assigned to a tape drive or a disk drive. The VISION:Builder symbolic unit for this file is SYS028; the DTF name is M4CHK2. When alternating checkpoints are written, each checkpoint file is opened immediately before writing the checkpoint and closed immediately after writing it. Only one checkpoint is stored on each of the two checkpoint files unless using the X specification.

Sort checkpoint option? (13)

Specifies that the CKPT parameter is added to the sort control statement images.

Entry	Result
Blank or N	Standard sort control statement images.
Y	The CKPT parameter is added to the SORT statement in the set of sort control statement images produced by VISION:Builder.

Note: Must be blank for IMS checkpoint/restart.

Time interval (14-16)

Specifies the time interval (in minutes or in seconds) between automatic checkpoints. Specifications should be right-aligned.

Entry	Result
1–999	Intervals, in minutes, at which checkpoints are taken. Checkpoint records are written at these intervals (minutes) unless a checkpoint occurs due to another specified checkpoint option.
nnS	Intervals, in seconds, at which checkpoints are taken. Checkpoint records are written at these intervals (seconds) unless a checkpoint occurs due to another specified checkpoint option. The “nn” value is the number of seconds from 01–99. The “S” indicates that the entry represents seconds.

Note: After the time interval has expired, the actual checkpoint is not written until just prior to reading the next master file record.

Record interval on file (count, control file entries) (17- 30)

Specifies a record count interval between automatic checkpoints, as well as the file used for the intervals.

Note: If you use this specification, both the count and control file specifications must also be completed.

Count (17-22)

Specifies the number of records processed before the checkpoint, but only in the cases of read-only and/or newly inserted records.

For existing records being updated or deleted, both record actions (get and replace or get and delete) are considered in the counting process and as such, two internal counts are taken for one record. Thus, the count is reached at an earlier point and the checkpoint is taken when approximately half the records specified by the count are processed. See general Rule D.

Processing Action	Internal Counts Taken
Read-only existing record	1
Update existing record	2
Delete existing record	2
Insert a new record	1

Entry	Result
1–999999	Intervals at which checkpoints are written. VISION:Builder writes checkpoints at this interval unless a checkpoint occurs due to another specified checkpoint option. Specifications should be right-aligned.

IMS checkpoints and counts are taken only when no new IMS or VSAM database is created (using M4NEW or M4SUBFn), when no one-step sort or report output is specified, when no program data definition changes are made before restart, when no packed files are used, and when VSAM files are used only as indexed coordinated files.

If user I/O, GDBI, or GSI is used, you must reposition your own files after each IMS checkpoint call. In this case, a successful restart is also dependent upon user coding.

If you use the record interval on file specification, both the count and control file specifications must be completed.

Control file (23-30)

Specifies the DTF/ddname of the file that controls the count. See general Rule A.

Operating System	Name of File that Determines the Record Interval for Controlling Checkpoints
OS/390, IMS, CMS	ddname.
VSE	DTF name.

Any input or output file can be used, except IMS checkpoint files.

For IMS checkpoint calls, the control file entry must be M4OLD.

If you use the record interval on file specification, both the count and control file specifications must also be completed.

Operator control? (31)

Operator can request checkpoints.

Entry	Result
Blank or N	Operator is not allowed to request checkpoints during processing.
Y	Operator is allowed to request checkpoints.

Not valid under IMS. Under OS/390, leave blank or enter N to prevent the operator from requesting checkpoints during processing.

If you enter Y, the operator must request all checkpoints by replying to the VISION:Builder message:

```
REPLY CP OR TERM WHEN CHECKPOINT IS REQUIRED.
```

VISION:Builder will be unable to write a checkpoint until the operator enters CP in response to the message. If the operator enters TERM, VISION:Builder will write a checkpoint, terminate the run, and print the file statistics.

This option is useful when performing update-in-place in order to avoid updating the same record twice after starting from a checkpoint.

If a checkpoint is required because another checkpoint option has been specified, VISION:Builder will output message MK4CS05 to the operator specifying that a checkpoint is requested by the program (see the *Messages and Codes* guide). At that time, the operator can enter CP in response to the original message:

```
REPLY CP OR TERM WHEN CHECKPOINT IS REQUIRED.
```

After the checkpoint is taken, this message will be issued again to allow the operator to request further checkpoints.

End of volume on file(s) (32-39)

Specifies that a checkpoint be written when end of volume occurs on a file.

Operating System	Name of File on Which the Checkpoint is to be Taken
OS/390	ddname.
All	ALL (checkpoint is written when EOV occurs on any legally named file).

Not valid under IMS or VSE.

M4INPUT, M4LIST, M4LIB, and M4SORT are not allowed.

IMS checkpoint ID prefix (40-46)

Entry	Result
Blank	The operating system assigns an 8-position checkpoint ID.
1–7 alphanumeric characters	Specifies a user prefix for the checkpoint ID. Any unassigned positions are assigned by the operating system. This specification must be left-aligned.

Note: This entry must be blank if a Y is entered in [47].

DB2 commit only? (47)

Entry	Result
Blank or N	VISION:Builder performs normal checkpoint processing. If relational files are present in the run, an implicit or explicit DB2 COMMIT statement is performed at the time of the checkpoint.
Y	VISION:Builder issues a DB2 COMMIT statement instead of a checkpoint. No restart is possible when this option is specified.

Note: This entry must be blank or N if there is an entry in [40-46].

CR Statement - Library Maintenance

Cataloged requests are executed by including a "Use Cataloged Request" (CR) statement in a run.

Position	Entry
01-08	Request/request group name
09-10	Statement type
11-16	Report date
17-44	Requestor name, telephone/ext., division/dept.

Request/request group name (01-08)

Required entry

Name of the request or request group as cataloged in the library. See general Rule A. Enter the 8-character request or request group name.

Statement type (09-10)

Required entry

Identifies a cataloged request statement. Enter CR.

Report date (11-16)

The date (if requested) is printed in the upper left corner of every page on the report. Date location can be overridden by an entry on the En statement. Unless you enter a specific date, the date printed is the date the report is run. Leave blank for preselection requests.

Entry	Result
Blank	No date.
XXXXXX	XX/XX/XX (user specified value).
TODAY	MM/DD/YY (01/15/95).
TODAYX	MM/DD/YYYY (01/15/2000).
DATE	MMM DD, YYYY (JAN 15, 1995).
ISDATE	YYYY-MM-DD (1995-01-15 ISO format).
JULANX	YYYY.DDD (2000.015).
JULIAN	YY.DDD (95.015).
SAME	Same date as originally cataloged.

Any of these entries except SAME overrides the ER report date. In installations where the date format has been changed with M4PARAMS, the new format is used.

Requestor name, telephone/ext., division/dept. (17-44)

Identifies the recipient of the report. Any information entered here prints on a page preceding the report.

Enter the word SAME in [17-20] followed by blanks in [21-44] and you will get the requestor information as originally cataloged.

CT Statement - Library Maintenance

Position	Entry
01-08	Request, request group, or definition name
09-10	Statement type
11-13	Operation
14-61	Request, group, or definition name
62	Continuation character

The CT statement is used to perform the following library maintenance functions:

- Save or delete requests in the common library (singly or in groups).
- List library contents, definition glossaries, or cataloged requests.
- Retrieve definitions and cataloged requests in source statement form.
- Move cataloged items from one common library to another.

The source statement retrieval (SSR) capability allows you to retrieve, in source statement form:

- File, array, and table definitions.
- Request and transaction groups.
- Requests.

These statements can be copied to an output file (M4SSOUT).

The source statement retrieval is a distinct run type indicated by an entry in position [38] of the RC statement. You cannot perform SSR operations in other types of runs (such as dictionary definition and processing).

- Input files are M4LIB and M4INPUT. Output files are M4LIST and M4SSOUT. No other files can be included.
- In the SSR run, the run control group consists of one RC statement and, if needed, one or more RP statements. This run control group must be the first set of statements in the run.
- A valid entry in the source statement out specification on the RC statement defines a source statement retrieval run. For VSE, the entry also specifies labeling conventions for the source statement output file (M4SSOUT).

Except for RP and CT statements, all other M4INPUT records of any type (for example, JCL or SYSTEM commands) that follow the RC statement are copied directly to the source statement output file (M4SSOUT) with no other action performed on them. CT statements and continuation CT statements containing any of the four retrieval operation codes are not copied to the source statement output file (M4SSOUT) but rather define the definition items to be retrieved and output in source statement form before any subsequent input stream statements are accessed.

All statements, except the run control group and the CT statements that specify SSR operations, are copied from M4INPUT directly to M4SSOUT. Thus, you can create a data set on M4SSOUT consisting of M4LIB source statements, other VISION:Builder statements, JCL, and data that can be used in a later run.

When a CT statement with an SSR operation is encountered, VISION:Builder retrieves the specified source statements from M4LIB and outputs them to M4SSOUT before any subsequent M4INPUT statements are accessed.

M4SSOUT is an unblocked file containing 80-character fixed length records. OS/390 users can override block size. For VSE systems, the last record output to M4SSOUT is a /* statement. The logical unit assignment under VSE is SYS037. The file type under OLX is M4SSOUT. Own-code hooks 10, 20, 21, 63, and 91 are supported under SSR runs.

The following list notes the various library maintenance operations and the type of run where each takes place. The types of runs are mutually exclusive.

Operation	Type of Run
REP, SAV, INS, DEL, LST, DMP	Processing.
LFN, LFG, LFV, LAN, LAG, LGN, ❹ LGG, ❹ LTN, LTG, LIN, LDN, LDG	Library definition.
RAD, RFD, RTD, RGD, ❹ RRG, EOF, RDB	Source statement retrieval.

Request, request group, or definition name (01-08)

Identifies the request or request group that is cataloged; otherwise, identifies for source statement retrieval a new name for a definition, request, or request group. See general Rule A. All requests making up the request group must pertain to the same master file definition. This entry must be different from any of the request names making up the request group.

Operation	Entry
INS	8-character name.
DEL, SAV, RAD, RFD, RTD, RRG, RGD,[4] RDB	8-character name or blank.
REP, LST, DMP, LFN, LFG, LFV, LAN, LAG, LGN,[4] LGG,[4] LTN, LTG, EOF, LIN, LDN, LDG	Blank.

Statement type (09-10)

Required entry

Identifies a library maintenance statement. Enter CT.

Operation (11-13)

Required entry

Specifies which library maintenance operation is to be performed: cataloging, catalog listing, library listing, or retrieval. Except for cataloging with catalog listing in a processing run, these operations cannot be mixed in a single run.

Catalog Operations

To *save a request or group*, enter SAV (valid only in a request processing run). The SAV operation requires an uncataloged request group name in name of request group to create a request group. There are three variations of the SAV operation:

- All requests in the run are cataloged in the request group if request group name is the only specification entered. The order of requests in the group is the same as the order in the run.
- Individual requests in the run are cataloged without creating a request group if their names are entered in request, group, or definition name and the name of request group is left blank.
- Selected requests in the run and/or requests and/or request groups in the library are cataloged in the request group if their names are entered in request, group, or definition name [14-61]. Requests are cataloged in the request group in the sequence specified on the CT statement.

To *insert a request* into a group, enter INS (valid only in a request processing run). The INS operation inserts a request or request group into the group named in the request or definition name [01-08].

- In name1 [14-21], enter the request or group name after which the insertion is to be made, followed in name2 [22-29] by the name of the request or group to be inserted.
- If name1 is blank, the request in name2 is inserted as the first request in the group.

To *replace a request* in all groups in which it occurs, enter REP (valid only in a request processing run). The cataloged request entered in request, group, or definition name [14-61] will be replaced with a request of the same name included in this run.

- The replacement is performed for all groups containing that request name.
- This operation applies to single requests only (that is, a request group cannot be replaced by another request group).
- Name of request group must be blank.

To *delete a request or a group*, enter DEL (valid only in a request processing run). The DEL operation performs three functions.

- Delete an entire request group. Enter the group name in name of request group. Leave all other specifications blank. Individual requests are not deleted when the group name is deleted because they may belong to other request groups as well.
- Delete one or more requests from a group. Enter the group name in name of request group and the request name(s) in request, group, or definition name [14-61].
- Delete a request from the library and all groups to which it may belong. Enter the request name in request, group, or definition name [14-61]. Leave all other specifications blank.

See Catalog Operations on page 4-19 for more information.

Catalog Listing Operations

To *list all requests and group names*, enter LST (valid only in a request processing run). The LST operation produces a listing of the names of all requests and request groups in the library.

- Leave all other specifications blank.
- Only one LST statement is effective per run although multiple LST statements can be specified.

To *list contents of all requests*, enter DMP (valid only during a request processing run). The DMP operation produces a list of the source statements making up each cataloged request or group specified in request, group, or definition name [14-61].

- If no entry is made in request, group, or definition name [14-61], a formatted listing of source statements for every request in the library is produced.

Library Listing Operations

To *list file definition names*, enter LFN (valid only in definition/maintenance runs). The LFN operation lists all file definition names in the library along with the names of their corresponding transaction group definitions. When the LFN operation is specified, any other entry on the CT statement is ignored.

To *list file definition glossaries*, enter LFG (valid only in definition/maintenance runs). The LFG operation lists all or selected file definition glossaries and their corresponding transaction group definition names.

- If no other entries are made on the CT statement, glossaries are listed for all file definitions in the library.
- If any file names are specified in name1–name6, only glossaries for those file definitions are listed.
- All file definition names in the library are listed before the glossaries when this operation is specified.
- The glossary listing format depends on the glossary type specified in the original file definition.

To *list file definition view glossaries*, enter LFV (valid only in definition/maintenance runs). The LFV operation lists file definition view glossaries.

- Name1–name6 specify the desired definition view glossaries to be listed.
- If left blank, all definition view glossaries will be listed.

In either case, file definitions on fields within them that cannot be used by VISION:Builder will not be listed. A message will be issued if a glossary is omitted, and the glossary heading will indicate whether or not fields have been suppressed.

All definition names specified along with their associated transaction groups will also be listed prior to the glossaries when this operation is specified. Individual glossary format will be the type specified in the original file definition.

To *list array definition names*, enter LAN (valid only in definition/maintenance runs). The LAN operation lists all array definition names in the library. When the LAN operation is specified, any other entry on the CT statement is ignored.

To *list array definition glossaries*, enter LAG (valid only in definition/maintenance runs). The LAG operation lists all or selected array definition glossaries.

- If no other entries are made on the CT statement, glossaries are listed for all array definitions in the library.
- If any array names are specified in name1–name6, only the glossaries for those definitions are listed.
- The glossary listing format depends on the glossary type specified in the original array definition.

To *list transaction group names*,[4] enter LGN (valid only in definition/maintenance runs). The LGN operation lists all transaction group definition names with their corresponding file definition names.

- When the LGN operation is specified, any other entry on the CT statement is ignored.

To *list transaction group glossaries*,[4] enter LGG (valid only in definition/maintenance runs). The LGG operation lists transaction group definition glossaries.

- If only the LGG entry is made on the CT statement, glossaries are listed for all the transaction group definitions in the library.
- To list specific transaction group glossaries, enter the transaction group names in name1–name6.
- All transaction group definition names in the library are listed before the glossaries when this operation is specified.

To *list table definition names*, enter LTN (valid only in definition/maintenance runs). The LTN operation lists all table definition names in the library.

- When the LTN operation is specified, any other entry on the CT statement is ignored.

To *list table definition glossaries*, enter LTG (valid only in definition/maintenance runs). The LTG operation lists all or selected table definitions.

- If only the LTG entry is made on the CT statement, glossaries are listed for all the table definitions in the library.
- To list glossaries for specific tables, enter the table names in name1–name6.
- The glossary listing depends on the glossary type specified in the original table definition.

To *list a library contents report*, enter LIN (valid only in definition/maintenance runs). The LIN operation lists all cataloged item names with item tracking information.

- When an LIN operation is specified, any other entry on the CT statement is ignored.
- The report is sequenced by item type and a page eject occurs when the item type changes.

Each report line includes:

- Item name.
- The date and time the item was created.
- The creator's ID (installation dependent).
- The date and time the item was last updated.
- The updater's ID (installation dependent).
- The date and time the item was last used.
- The expiration date/retention period of the item.

To *list logical data view or run data group names*, enter LDN (valid only in definition/maintenance runs). The LDN operation lists all logical data view or run data group names in the library.

- When the LDN operation is specified, any other entry on the CT statement is ignored.

To list *logical data view or run data group glossaries*, enter LDG (valid only in definition/maintenance runs). The LDG operation lists all or selected logical data view or run data group glossaries.

- If no other entries are made on the CT statement, glossaries are listed for all logical data view definitions in the library.
- If any names are specified in name1–name6, only glossaries for those logical data views are listed.
- All logical data view names in the library are listed before the glossaries when this operation is specified.
- The glossary listing format depends on the glossary type specified in the original logical data view.

Retrieval Operations

To *retrieve file definitions*, enter RFD (valid only in source statement retrieval runs). The RFD operation retrieves all or selected file definitions.

- If no other entries are made on the CT statement, all file definitions are retrieved.
- If any names are specified in name1–name6, only the definitions for those files are retrieved.
- A retrieved file definition can be renamed by specifying the new name in request or definition name [01-08] and the file to be renamed in name1 [14-21].
- When renaming a file, the rest of the CT statement must be kept blank.

To *retrieve table definitions*, enter RTD (valid only in source statement retrieval runs). The RTD operation retrieves all or selected table definitions.

- If no other entries are made on the CT statement, all table definitions are retrieved.
- If any names are specified in name1–name6, only the definitions for those tables will be retrieved.
- A retrieved table definition can be renamed by specifying the new name in request or definition name [01-08] and the name of the table to be renamed in name1 [14-21].
- When renaming a table, the rest of the CT statement must remain blank.

To *retrieve array definitions*, enter RAD (valid only in source statement retrieval runs in name1–name6). The RAD operation retrieves the array definitions specified or, if none are specified, retrieves all of them.

- It is also used to output an array definition to M4SSOUT under a revised name.

To *retrieve transaction group definitions*, ❹ enter RGD (valid only in source statement retrieval runs). The RGD operation retrieves all or selected transaction groups.

- Name1 [14-21] will contain the name of the group to be retrieved; name2 [22-29] will contain the name of the master file for the definition.
- If both the group name and the master file name are blank, all transaction groups will be retrieved.
- If only the master file name is specified, all transaction groups for that file are retrieved.
- Request or definition name [01-08] can contain a name that will be used to rename the retrieved transaction group specified in name1 [14-21].
- Name3 [30-37] can contain a name that will be used to rename the master file entry on the retrieved TD statement.
- When name1 [14-21] specifies the name of a group to be retrieved, the name of the master file name2 [22-29] must be specified.

To *retrieve request/request groups*, enter RRG (valid only in source statement retrieval runs). The RRG operation retrieves all or selected requests.

- If no other entries are made, all requests and request groups are retrieved.
- If any names are specified in name1–name6, only the source code for those groups is retrieved.
- A retrieved request can be renamed by specifying the new names in request or definition name [01-08] and the name of the request in name1 [14-21].

To *retrieve logical data view or run data group definitions*, enter RDB (valid only in source statement retrieval runs). The RDB operation retrieves all or selected logical data view or run data group definitions.

- If no other entries are made on the CT statement, all logical data view definitions are retrieved.
- If any names are specified in name1–name6, only the definitions for those logical data views are retrieved.
- A retrieved definition can be renamed by specifying the new name in request or definition name [01-08] and the file to be renamed in name1 [14-21].
- When renaming a file, the rest of the CT statement must remain blank.

The *EOF operation* is valid only in source statement retrieval runs and causes a /* statement to be written to the M4SSOUT file. Enter EOF.

Request, group, or definition name (14-61)

See general Rule A.

Name1 (14-21)
Name2 (22-29)
Name3 (30-37)
Name4 (38-45)
Name5 (46-53)
Name6 (54-61)

For SAV, DEL, INS, and REP cataloging operations, enter up to six names on one line; use a second line when more than six names are required for catalog maintenance.

- Names must be request or request group names included in this run or already cataloged.
- Enter the first name in name1, the second in name2, and so on. The only exception is an INS operation where name1 is blank and the entry in name2 is to be inserted at the beginning of a request group.
- As noted earlier, request and request group names must be unique.
- When cataloging a single request name that will not be saved under a group, enter the name in request, group, or definition name leaving name of request group blank.
- For DMP, LFN, LFG, LGN,[4] LGG,[4] LTN, LTG, LAN, LIN, LFV, and LAG listing operations, enter up to six names on one line; use a second line when more than six names are required.

Continuation character (62)

Enter any nonblank character to indicate that the following line contains names in request, group, or definition name for the operation in this line.

- Leave name of request group and operation blank in a continuation line.
- More than one continuation line is allowed; all but the last line must contain a nonblank character in this entry.
- A continuation character cannot be used for the LST, LFN, LGN,[4] LTN, LAN, RGD,[4] and EOF operations.

DB Statement - Run Data Group

The DB statement specifies the name of the data group or logical data view. You must have one DB statement for each run data group or logical data view you define.

Position	Entry
01-08	Data group name
09-10	Statement type
19	Delete?
20	Glossary

The run data group (logical data view) conveniently catalogs your run control information. Run data groups are VISION:Builder run control statements saved in the common library for use during processing. The following statement types can be cataloged in a run data group:

DB, RF, LB } Minimum Requirements
RT
RP
RG
OC
CP
PA
RA
WH
CR

AA and IT statements can be included in a run data group but are not cataloged.

Data group name (01-08)

Required entry

This is the name of the data group being defined. See general Rule A.

Statement type (09-10)

Required entry

This entry identifies this as the DB statement. Enter DB.

Delete? (19)

Entry	Result
Blank or N	Definition is added to the library.
Y	Definition is deleted from the library.
R	Definition is replaced if it is already cataloged; otherwise, it is added to the library.

Glossary (20)

Selects a glossary listing of the data group (with aliases) described by this definition. To request glossaries for previously defined data groups, use a DB statement with only the data group name and glossary type entered.

Entry	Result
Blank or A	Format 1 alphabetic list by field name within each segment.
B	Format 1 sequential listing by location of fields within each segment.
1	Format 1 abbreviated listing, alphabetic by field name within each segment.
X	Format 2 glossary listing.
N	No glossary.

D1 Statement - File Definition

The D1 statement is used to specify the first 40 characters of the field description. Field descriptions can be up to 70 characters long.

Position	Entry
01-08	File name
09-10	Form code
11-18	Short field name
19	Delete?
30-69	Field description

File name (01-08)

Required entry

Identifies the file containing the field. Enter the 8-character name as entered on the FD and L0 statements.

Form code (09-10)

Required entry

Identifies a code defining the attributes of a field in the file definition. Enter D1.

Short field name (11-18)

Enter the short field name from the same columns on the L0 statement for this field.

Delete? (19)

Deletes field description.

Entry	Result
Blank or N	Description added to field.
Y	Description deleted from field.

Field description (30-69)

Enter the first 40 characters of the field description assigned to this field. If you need more than 40 characters, continue the description on the D2 statement.

D2 Statement - File Definition

The D2 statement is used to specify the last 30 characters (columns 30-59) of the field description. The total D1/D2 field descriptions can be up to 70 characters long.

Position	Entry
01-08	File name
09-10	Form code
11-18	Short field name
19	Delete?
30-59	Field description

File name (01-08)

Required entry

Identifies the file containing the field. Enter the 8-character name as entered on the FD and L0 statements.

Form code (09-10)

Required entry

Identifies a code defining the attributes of a field in the file definition. Enter D2.

Short field name (11-18)

Enter the short field name from the same columns on the L0 statement for this field.

Delete? (19)

Deletes field description.

Entry	Result
Blank or N	Description added to field.
Y	Description deleted from field.

Field description (30-59)

Enter the last 30 characters of the field description assigned to this field.

En Statement - Output Specification

The output specification statement (En statement) formats reports and specifies subfiles, alternate report files, report summary files, and alternate M4LIST files to be output.

If the En statement is used in a report that produces a graph, see Chapter 5, *Graphic Fixed Format Statement Listing* for the En statement specifications for graphics output.

Position	Entry
01-08	Request name
09-10	Statement type
11	Summary report only
12	Vertical spacing
13	Print 8 lines per inch?
14-16	Width of page
17-19	Height of page
20	Number of repeated images
21	Special forms
22-23	Maximum number of lines per page
24-27	Maximum number of pages
28	Page title at bottom of page?
29	Column heading type
30	Column heading position

Position	Entry
31-32	Date position
33-34	Page number position
35-38	Start page numbers at
39	Line numbers
40	Labels on summary lines
41-56	File entries
41-42	Report handling
43	Entire record selection
44-51	DTF/ddname or file name
52-55	Subfile blocking factor
56	Subfile format
57	Column heading character
58	Empty field control
59	Autogrand?
60	Multi image single title?
61	Repeating subtitles?
62	Invalid summary indicator
63	Suppress NDS report?
72	Continue

Request name (01-08)

Required entry

Identifies this statement and others accompanying it (if any) as one request. See general Rule A.

Enter the same request name for all statements in the request.

Statement type (09-10)

Required entry

Enter an E followed by a number (n) from 1–9 to relate this statement to a specific report. The number must be the same on all En, Fn, Kn, Rn, Pn, Tn, and Wn statements in a report output statement group.

Note: Each request can generate up to nine reports, each in a different format.

Summary report only (11)

Summaries are output without detail printing.

Entry	Report Produced
Blank or N	Detail and summary information.
Y	Summary only with double-spacing between summary levels.
S	Summary only with single-spacing between summary levels and double-spacing after the level one summary.

Vertical spacing (12)

Specifies the spacing down the report page.

Entry	Result
Blank or 1	Single-spacing.
2 to 9	1–8 blank lines between each detail line.

Note: One extra line is skipped after each set of summaries.

Print 8 lines per inch? (13)

The system computes the page size to print 8 lines per inch instead of 6.

Entry	Result
Blank or N	6 lines/inch.
Y	8 lines/inch.

Width of page (14-16)

Specifies the width of the printed page.

Leave blank for preselection requests.

Enter, left-aligned, one of the following codes to select the corresponding page width.

Entry	Maximum Number of Characters Across Page	Width in Inches
Blank	Installation Standard	M4PARAMS – Default Page Width
A	110	11.0
B	85	8.5

Entry	Maximum Number of Characters Across Page	Width in Inches
C	73	7.3
D	50	5.0
E	35	3.5

Fields extending beyond the specified width print on the next (folded) line.

For other page widths, enter a right-aligned number from 1 through the installation standard as set by M4PARAMS equal to the number of characters to print across the page.

Height of page (17-19)

Specifies the height of the printed page, including margins, in single lines.

Leave blank for preselection requests. 

Enter, left-aligned, one of the following codes to select the corresponding page height.

		Maximum Number of Lines	
Entry	Height in Inches	6 Lines/Inch	8 Lines/Inch
Blank	Installation Standard	Installation Standard	Installation Standard
A	8.50	51	68
B	11.00	66	88
C	3.25	19	26
D	1.00	6	8
E	22.00	132	176

Note: For other page heights, enter a right-aligned number from 2 to 255 equal to the number of lines to print on each page.

VISION:Builder computes the default height in lines as follows:

- If the page height in inches is greater than one inch, the page height in lines equals user specified page height minus one inch.
- If less than one inch or if formatted reporting is being used, the page height in lines equals user specified page height minus one line.

Number of repeated images (20)

Specifies the number of images (number "up") printed across the page (depending on page width).

Entry	Result
Blank	Installation standard page output.
Any number	Number of horizontal image areas to be printed.

Note: Page width times the number of images cannot exceed the installation standard page width in M4PARAMS. (Width of page [14-16] must be specified as width of each image.)

Special forms (21)

Requests forms other than installation standard paper. This specification has no effect on systems using spooled output.

Entry	Result
Blank or N	Installation standard forms.
Y or S	Special forms.

Note: When Y or S is specified, a message prints to alert the operator to mount special forms.

Maximum number of lines per page (22-23)

Restricts printing to part of a page (limiting the number of lines printed per page).

Entry	Result
Blank	Maximum number of lines possible depending upon page height.
2 to 99	Number of lines printed on each page of either detail or summary report (including blank lines, column headings, totals, and summaries). Cannot exceed height of page minus one.
NM	No margin will be subtracted from the page height for the report, and the full page height will be used.

Note: This entry is not applicable for formatted reporting.

Maximum number of pages (24-27)

Limits the total number of printed pages for a report but does not limit the data selected for output.

Entry	Result
Blank	All selected data is printed.
1–9999	Only 1–9999 pages are printed.

Note: Data that would normally be printed after the number of pages indicated is not printed.

Page title at bottom of page? (28)

Specifies the position for printing the report title specified on the Tn statement.

Entry	Title Position
Blank or N	Top of page.
Y	Bottom of page.

Note: This entry does not apply to formatted reporting.

Column heading type (29)

Allows using field names for column headings or eliminating headings.

Entry	Column Headings
Blank	As shown on file definition or temporary field definition.
F	Field names (Rn statement) become the column headings.
X	No column headings.

If F is specified, trailing blanks are eliminated and the field name is centered over the column.

Not applicable for formatted reporting or for formatted sectional reporting if the column heading section is overridden.

Column headings begin two lines below the title.

The detail line begins two lines below the column heading (if any).

Column heading position (30)

Specifies the placement of the column headings on the report.

Entry	Column Heading Position (Centered Over or Under Columns of Data)
Blank or T	Top of page.
B	Below detail items and summaries.

If B is specified and a Y is entered in page title at bottom of page [28], the column headings print above the page title.

When column headings are placed across the page, VISION:Builder allocates space for the following items per field:

- The data in the column.
- The column headings themselves.
- The spaces before the column [14-15] on the Rn statements. The data and column headings will overlap across the page; the spaces before the columns will not.

Date position (31-32)

Specifies the placement of the date on the report.

Entry	Date Position
Blank or UL	Upper-left corner of page.
UR	Upper-right corner of page.
LL	Lower-left corner of page.
LR	Lower-right corner of page.
MT	Middle top of page.
MB	Middle bottom of page.
ND	No date.

If date position is specified, you must also specify the report date [11-16] on the ER statement or the date does not print.

Not applicable for formatted reporting.

Page number position (33-34)

Specifies the placement of the page number on the report.

Entry	Page Number Position
Blank or UR	Upper-right corner of page.
UL	Upper-left corner of page.
LL	Lower-left corner of page.
LR	Lower-right corner of page.
MT	Middle top of page.
MB	Middle bottom of page.
NP	No page numbers on this report.

If both page number and date are specified for the same place, only the page number prints.

Not applicable for formatted or formatted sectional reporting.

Start page numbers at (35-38)

Specifies the page number for the first report page. Maximum starting number allowed is 9,999.

Entry	Result
Blank or 1	Page numbering begins at 1.
2 to 9999	Page numbering begins at the number specified (useful for special or partial reports).
PAGE	Resets the page number to 1 each time a level 1 page subtitle break occurs.

The preface page (Pn statement) is not numbered.

Not applicable for formatted reporting.

Line numbers (39)

Specifies and positions line numbers for detail reports only. Lines are counted and numbered by logical detail lines rather than actual printed lines. Therefore, if the end line is used (Rn statement [25]) or if the line is folded, it is counted as only one line.

Entry	Result
Blank or N	No line numbers.
L	Prints line numbers at left side of page beginning at 1.
R or Y	Prints line numbers at right side of page beginning at 1.
B	Prints line numbers at both sides of page beginning at 1.

If line numbers are specified, six positions are reserved for them. If you also specify width of page [14-16], you must specify a page width large enough to accommodate both the detail line and the line number.

On folded lines, left line numbers print on the left of the first line, right line numbers print on the right of the last of the folded lines.

Line numbers are reset whenever page numbers are reset.

Labels on summary lines (40)

Positions or suppresses the 14 spaces of the summary descriptions (labels) when summaries are requested (Rn statement [31-36] and [41-52]).

Entry	Result
Blank	The 14 spaces are reserved at the extreme left for the summary labels.
L	Columns of report data begin at the left of the report.
X	No summary labels. If summaries are requested, they print left-aligned without descriptions.

Of the 14 spaces, [01-08] are reserved for the name of the control break field and [10-14] are for the type of summary.

If a summary is requested on columns in spaces [1-14], the label will be overlaid by the summary. Therefore, position your data so that those with summaries begin no earlier than position [15].

Not applicable for formatted reporting or formatted sectional reporting when the summary section is overridden.

File entries (41-56)

File entries (report handling, entire record selection, DTF/ddname or file name, subfile blocking factor, subfile format) are used to identify a subfile and its attributes. Report handling and DTF/ddname or file name are used to identify the alternate M4LIST file, report summary file, and alternate report file.

Note: Multiple reports may refer to the same subfile if it is not a relational table or IMS database.

Report handling (41-42)

Specifies whether reports, alternate report files, alternate M4LIST files, subfiles, file definitions, and report summary files are generated or alternate report output methods are used.

Entry	Result
Blank	Report and subfiles are generated if other file entries are specified.
NR	No report. Other entries: DTF/ddname or file name, subfile blocking factor, subfile format.
RF	Report placed on a separate report file. Other required entries: DTF/ddname or file name and an RF statement that specifies the alternate report file.
DN	Subfile and subfile file definition created. No report.
DR	Subfile, subfile file definition, and report are generated.
AL	This report and the report START, END, NO DATA, INCOMPLETE SUMMARY, NUMBER OF LINES OF PRINT, as well as NUMBER OF MISSING/INVALID/OVERFLOW FIELD messages are printed on the alternate M4LIST. All signons, source statements, messages, and statistics are printed on M4LIST.
SD	A report summary file and a report are created. A file definition is generated.
SF	A report summary file and a report are created. No file definition is generated.
TD	Tab-delimited output method is used.
CD	Comma-delimited (CSV) output method is used.
PT	Plain text output method is used.
HT	HTML document output method is used.
DD	Extended subfile and subfile file definition created. No report.

Entry	Result
DO	Extended subfile created. No report.
RD	Report is output as a raw data file. Subfile file definition created.
RO	Report is output as a raw data file.

Report summary files, reports, subfiles, generated file definitions, alternate M4LIST files, and alternate report files can be specified simultaneously. However, you cannot suppress a report (DN or NR) and request a report summary file (SD or SF).

File definition statement images are written on the subfile specified on the RF statement having a C in position [53]. The file definition is generated with the name entered in DTF/ddname or file name [44-51].

Entire record selection (43)

Identifies special handling of selected input file records or data for subfiles.

Entry	Output of Entire Selected Records or Coordinated Records to Subfiles
Blank	The fields specified on the Rn statement produce a report or subfile.
O or 0	Output selected old master file records to a subfile.
N	Output selected new master file records to a subfile.
1–9	Output selected coordinated file records (1–9) or array records to a subfile.
X	Output selected transaction file records to a subfile during transaction request processing.❹
A, B, E, G, H, J, K, M, Q	Output selected array records to a subfile. Code must match the one listed on the RF statement.

- Rn statement sort sequence [27] and summary specifications [31–36] do not cause sorting or summarizing on subfiles.
- This entry must be blank if creating a relational table as a subfile.❶
- This entry must be blank if any of the following Report Handling entries are specified: SD, SF, TD, CD, CL, PT, HT.
- This entry cannot be blank when generating an IMS subfile.❶
- This entry must contain N for the dynamically deleted master segments entry on the RF statement [53] to have any effect on the subfile.
- If O or N, the format of the subfile is identical to that of the master file unless otherwise specified on the En statement. Selected master file records or the master file can be output only on a subfile.

  If a blocking factor and/or format are specified, they define the subfile format.

  If an entry in the subfile format changes the record format from V or U to F, the record length of the subfile is equal to the maximum input file record length.

If fields are specified on the Rn statement (in addition to an entry in this specification) and NR is not specified in report handling, the entire record is output to the subfile, and the report contains only the specified fields.

- If X is entered, output will be blocked as requested in the subfile blocking factor specification. The record format for code X is the same as that of the transaction file unless an entry is made in subfile format. This code is valid only in transaction requests.
- When an IMS subfile is output from a conventional input file, the file definition of the input file must have every segment named on LS statements that matches the DBD segment names. I

DTF/ddname or file name (44-51)

Identifies the subfile, alternate M4LIST file, relational table, report summary file, alternate report file, or alternate report output method file produced from this request. See general Rule A. (This name must also be entered on an RF statement if it is a subfile, relational table, or alternate report file.)

- If a file definition is created, it is identified by this name.
- The same subfile name can be used by more than one request, allowing output to the same subfile for various criteria. Whenever a subfile is produced from more than one request, all requests must have the same entries in the subfile blocking factor and subfile format specifications. If a subfile is required without a report, NR must be entered in report handling.
- The same subfile name cannot be used by more than one subfile output request if the subfile format is a DB2 table or IMS database. D I

Non-IMS files, alternate report files, and relational table generation

Enter a unique 8-character name (must match RF).

IMS files (subfiles only)

Enter a user defined DBD name (up to 8 characters and must match RF). I

Alternate M4LIST

Entry	Result
Blank	M4LIST1 is used as the alternate M4LIST DTF/ddname for the report.
Unique name up to 8 characters (7 for VSE)	This unique name is used as the alternate M4LIST DTF/ddname for this report. For VSE, SYS055 will be used as the SYS number for all alternate M4LISTs unless they are overridden by RP statements.

Note: An RF cannot be included for an alternate M4LIST; however, a DTF/DD statement must be included in the JCL.

Report summary files

Entry	Result
Unique name up to 8 characters (7 for VSE)	This unique name is used as the DTF/ddname for the report summary file generated by this report. For VSE, SYS056 is used as the SYS number for all report summary files unless overridden by RP statements.

- An RF statement cannot be included for a report summary file; however, a DTF/DD statement must be included in the JCL.
- Multiple reports cannot be output to the same report summary file. This name can be used by only one report.
- A report summary file cannot be specified in conjunction with the RPO option on the RP statement.

Alternate Report Output Method files

Entry	Result
Unique name up to 8 characters (7 characters for VSE)	This unique name is used as the DTF/ddname for the alternate report output method file generated by this report. For VSE, SYS057 is used as the SYS number for all report summary files unless overridden by RP statements.

- An RF statement cannot be included for an alternate report output method file; however, a DTF/DD statement must be included in the JCL.
- Multiple reports cannot be output to the same alternate report output method file. This name can be used by only one report.
- An alternate report output method file cannot be specified in conjunction with the RPO option on the RP statement.

Extended Subfiles (Report Handling DD or DO)

Adhere to the following standards:

- Enter a DBDNAME when the Subfile Format entry is D or H.
- Leave blank when the subfile format entry is S and the Report Handling entry is DO.
- Enter a file definition name for the created definition when the Subfile Format entry is S and the Report Handling entry is DD.
- Enter a DDname when the Subfile Format entry is any code except D, H, or S.

Note: When an Extended Subfile with Subfile Format has an entry of S, you must include a Wn statement.

Raw Data Output (Report Handling RD or RO)

Adhere to the following standards:

- Enter a DDname that the file name can be written to.

Note: Raw Data Output is similar to subfile output with Subfile Format V, except that the file is created during the report generation phase following any required sorting. The Subfile Format entry must be blank – the records are always written as variable length records (RECFM V).

Subfile blocking factor (52-55)

Describes the block size of physical records on the subfile as follows (if multiple requests refer to the same subfile, the attributes must be identical). For HTML output (the Report Handling entry is HT), [54-55] specifies HTML style for a report. See the *VISION:Builder for OS/390 Environment Guide* for information regarding the use of HTML styles.

The following information on the values that can be specified for the subfile blocking factor [52-55] entry are based on the value that is specified in the subfile format [56] entry.

HTML Report Output Style (54-55) (HT in Report Handling(41-42))

Entry	Result
Blank	Default style is used.
1-99	User-defined style is used. Requires setup (refer to the *VISION:Builder for OS/390 Environment Guide* for a description).

Variable blocked (V or blank in subfile format (56))

This value represents the buffer size and must be large enough to accommodate the largest record output. If the operating system supports system-determined block size and the file meets the criteria for this capability, this value is ignored and is determined by the operating system.

Entry	Entire Record Selection Entry?	Result
Blank	Yes	Value defaults to that specified in the input file.
Blank	No	Value defaults to installation-specified block size for M4REPO (see M4PARAMS listing in the *VISION:Builder for OS/390 Installation Guide* or the *VISION:Builder for VSE Installation Guide*).
1–9999		Buffer size.
1H–327H		Buffer size, interpreted as 100–32,700.
nnnK		Multiples (nnn) of 1024 bytes. If the value is greater than 31K, it is interpreted as variable blocked spanned record size.

Fixed blocked (F in subfile format (56))

If the operating system supports system-determined block size and the file meets the criteria for this capability, this value is ignored and is determined by the operating system (fixed block). Otherwise, fixed unblocked.

Entry	Result
Blank	Value defaults to that specified in the input file.
1–9999	Number of records per block.
1H–327H	Number of records per block, interpreted as 100–32,700.
nnnK	Multiples (nnn) of 1024 bytes (1K–31K).

- Value of 1 assumed if the entire record selection is specified.
- To create a fixed unblocked subfile under an operating system that supports system-determined block size, enter a 1 for subfile blocking factor or code RECFM=F on the subfile JCL statement.

Undefined (U in subfile format (56))

Entry	Entire Record Selection Entry?	Result
Blank	Yes	Value defaults to that specified for the input file.
Blank	No	Value defaults to 8 less than the installation specified block size for M4REPO.
1–9999		Buffer size.
1H–327H		Buffer size, interpreted as 100–32,700.
nnnK		Multiples (nnn) of 1024 bytes (1K–31K).

ISAM fixed blocked (I in subfile format (56))

Entry	Result
Blank	Value of 1 assumed if entire record selection is specified.
1–9999	Number of records per block.
1H–327H	Number of records per block, interpreted as 100–32,700.
nnnK	Multiples (nnn) of 1024 bytes (1K–31K).

Note: Unblocked ISAM files are not processed by VISION:Builder.

ISAM variable blocked (J in subfile format (56))

Not valid for VSE

Entry	Entire Record Selection Specified?	Result
Blank	Yes	The buffer size of the entire record selection file is used.
Blank	No	Report file buffer size is used.
1–9999		Buffer size.
1H–327H		Buffer size, interpreted as 100–32,700.
nnnK		Multiples (nnn) of 1024 bytes (1K–31K).

Packed (P in subfile format (56))

Entry	Entire Record Selection Specified?	Result
Blank	Yes	The buffer size of the entire record selection file is used.
Blank	No	Report file buffer size is used.
1–9999		Buffer size.
1H–327H		Buffer size, interpreted as 100–32,700.
nnnK		Multiples (nnn) of 1024 bytes. If the value is greater than 31K, it is interpreted as variable blocked spanned record size.

VSAM (E or K in subfile format (56))

Leave blank.

Relational table generation (S in subfile format (56))

Leave blank.

IMS (D or H in subfile format (56))

Leave blank.

Subfile format (56)

Describes the format of the subfile's physical records. If multiple requests refer to the same subfile, the attributes must be identical.

Entry	Record Type
Blank or V	Variable blocked or variable blocked spanned if subfile blocking factor is greater than 31K.
F	Fixed blocked.
U	Undefined.
I	ISAM fixed blocked.
J	Variable ISAM (not supported by all operating systems).
P	Packed (variable blocked or variable blocked spanned if subfile blocking factor is greater than 31K).
D	IMS subfiles.
K	Key sequenced VSAM.
E	Entry sequenced VSAM.
H	IMS HDAM subfiles.
S	Relational table generation.

- When using F format subfiles, data from different requests for the same subfile must have the same length (that is, for each request, the total length of the fields in the Rn statement must be the same).
- For DL/I files, if an automatic subfile definition is requested, the record format is H.
- This entry must be blank for a report summary file, which is always created as a variable blocked file.
- This entry must be blank for alternate report output method files.

Column heading character (57)

Allows you to override the installation standard column heading borders. See M4PARAMS in the *VISION:Builder for OS/390 Installation Guide* or the *VISION:Builder for VSE Installation Guide*.

Entry	Border
Blank	Installation standard.
B	Blank lines replace regular borders.
N	No borders (blank lines are not output in place of borders).
Any other character	The character specified is used for column heading borders instead of the installation standard.

Empty field control (58)

Specifies the handling of "empty" fields in summaries. (An "empty" character string field contains blanks; an "empty" numeric field contains zeroes.) Enter one of the following, depending on how you want your summaries computed.

Entry	Blank Character String Fields Included in Summaries?	Zero Numeric Fields Included in Summaries?
Blank	No	Yes
I	Yes	Yes
E	No	No

Autogrand? (59)

Specifies whether an automatic grand summary page is to be printed at the end of a specific report. If printed, it appears as a separate page at the end of the report.

Entry	Result
Blank	No change from M4PARAMS (see the *Installation Guide*) or from the selection on the RP statement.
N	No autogrand page is printed for this report.
Y	Autogrand page is printed for this report.

An entry of Y is not applicable for formatted or formatted sectional reporting.

If autogrand is in effect for a report with a report summary file specified, a record is written to the file at the grand level control break (end of the report).

Multi image single title? (60)

Allows a single title and page number to be output across the physical page when the report has been repeated multiple times on that same page. Preface pages are output one-up only.

Entry	Result
Blank or N	Titles and page numbers appear over each report image on the physical page.
Y	Single title and page number are output across the physical page.
P	Produces the same result as Y with one addition: if page subtitles exist, a change in the value causes a skip to the next physical page.

Note: Not applicable for formatted sectional reporting if the page title section is defined.

Repeating subtitles? (61)

Allows you to choose repeating or nonrepeating S-type subtitles by report. Refer to the output specification Rn statement for information on the format of the S-type subtitles.

Entry	Result
Blank	No change from M4PARAMS (see the *Installation Guide*).
Y	Subtitles repeat for this report if detail items continue on next pages, regardless of the M4PARAMS.
N	Subtitles do not repeat for this report regardless of the M4PARAMS.
P	Repeating subtitles print only on each new physical page of a multi-image report. This entry is ignored for single image reports.

Note: Not applicable for formatted reporting.

Invalid summary indicator (62)

Indicates whether or not report summaries are to be flagged on the report if they are incomplete. A summary is incomplete if one or more missing or invalid fields was involved in computing the summary. A valid summary value was able to be computed but it did not involve all the detail values.

Entry	Result
Blank	No flagging of incomplete summaries.
Any other character	Incomplete summaries will be flagged with this character on the report.

- A plus sign (+) prints if there is not enough room to print the complete summary value.
- An entry in this field applies only to the printing of a summary on the report. It does not affect an incomplete summary or its status indicator in the report summary record.

Suppress NDS report? (63)

Indicates whether the skeleton NDS (no data selected) report should print when there is no data for that report.

Entry	Result
Blank	Indicates that the system default, as specified in M4PARAMS, is used.
Y	Indicates that if an NDS condition occurs, the skeleton report is suppressed (overriding the system default).
N	Indicates that if an NDS condition occurs, the skeleton report is printed (overriding the system default).

Note: If a report summary file is requested, a record will be written only when the NDS report is printed.

Continue (72)

Indicates multiple En statements. Continuation is used for additional file entries [41-56] only.

Enter any nonblank character to specify continuation of the En statement.

ER Statement - Processing and Record Selection

The processing and record selection statement (ER statement) allows you to describe the request (that is, request name, report date, your name, and company location). It also gives you selection control options. The default formatting capabilities, and those specified on the ER statement, can be overridden on the En statement.

Position	Entry
01-08	Request name
09-10	Statement type
11-16	Report date
17-44	Requestor name, telephone/ext., and division/dept.
45-48	Maximum items selected
49	Selection control
50	Summary only report
51	Vertical spacing
52	Forms control
53	Width of page
54	Height of page
55	Line numbers?
56	Request type
57-64	Set name
65-69	Back branch control
70	Reinitialize temporary fields?
71	Abbreviated format

Request name (01-08)

Required entry

Identifies this statement and others accompanying it as one request. See general Rule A. Enter the same name for all statements in each request. The ER statement alone does not constitute a valid request.

Statement type (09-10)

Required entry

Identifies a processing and record selection header statement. Enter ER.

Report date (11-16)

The date (if requested) is printed in the upper-left corner of every report page. Date location can be overridden by an entry on the En statement.

Leave blank for preselection requests.

The following lists the default formats from M4PARAMS.

Entry	Result
Blank	No date.
XXXXXX	XX/XX/XX (user specified value).
TODAY	MM/DD/YY (01/15/95).
TODAYX	MM/DD/YYYY (01/15/2000).
DATE	MMM DD, YYYY (JAN 15, 1995).
ISDATE	YYYY-MM-DD (1995-01-15 ISO format).
JULANX	YYYY.DDD (2000.015).
JULIAN	YY.DDD (95.015).

Requestor name, telephone/ext., and division/dept. (17-44)

Prints on a separate page preceding the report and identifies recipient.

Leave blank for preselection requests.

Maximum items selected (45-48)

Limits the number of records selected or output in a request.

Leave blank for preselection requests.

Entry	Result
Blank	All items that meet the specified criteria are selected.
1–9999	Maximum number of items to be selected in the request. For multiple outputs in a request, each output record is counted in the maximum number, regardless of the number of Rn sets or Rn statements used or files output. Where no reports are output, each positive exit from the request is counted in the maximum; a record with lower level segments may, due to looping, positively exit the request more than once. A positive exit is one that results from a GO or GS branch or from simply "falling through" to the end of the request (an NS branch out of the request is not a positive exit).

Selection control (49)

Controls processing in repeated segments.

Leave blank for preselection requests.

Entry	Result
Blank or N	All occurrences meeting the specifications listed on the PR statements will be selected.
Y	Only the first occurrence within each record that meets the specifications listed on the PR statements will be selected.

Report format

These entries describe how the report will look when printed. If En statements are included, the report format entries (including blanks) on the En statements override the report format entries made on the ER statement.

Summary only report (50)

Lists summaries without detail lines.

Leave blank for preselection requests.

Entry	Result
Blank or N	Report with detail lines.
Y	Summaries only with double-spacing between summary levels.
S	Summaries only with single-spacing between summary levels and double-spacing after the level one summary.

Note: This entry must be blank when generating a graph.

Vertical spacing (51)

Specifies the spacing of the lines on a report page.

Leave blank for preselection requests.

Entry	Result
Blank or 1	Single-spacing.
2 to 9	2 to 9 spaces between detail lines on the report.

Note: This entry must be blank when generating a graph.

Forms control (52)

Alerts the operator to mount special forms.

Leave blank for preselection requests or when using spooled output.

Entry	Result
Blank or N	Installation standard forms.
S or Y	Special forms.

Width of page (53)

Specifies the width of the printed page.

Leave blank for preselection requests.

Code	Maximum Number of Characters Across Page
Blank	M4PARAMS Default Page Width
A	110
B	85
C	73
D	50
E	35

Height of page (54)

Specifies the height of the printed page.

Leave blank for preselection requests.

		Maximum Number of Lines	
Entry	Height in Inches	6 Lines/Inch	8 Lines/Inch
Blank	Installation Standard	Installation Standard	Installation Standard
A	8.5	51	68
B	11.0	66	88
C	3.25	19	26
D	1.00	6	8
E	22.00	132	176

Line numbers? (55)

Requests that detail lines be numbered.

Leave blank for preselection requests.

Entry	Result
Blank or N	No line numbers.
Y	Line numbers requested (printed consecutively from 1–999999 as needed).

Note: This entry can be used when producing a graph, but it applies to the data report.

Request type (56)

Specifies a transaction request type or subroutine or allows end of file processing for certain requests.

Entry	Points at Which the Request is Executed
Blank or N	During the passing of the master file.
A	During the passing of the master file, plus one more time when input files have reached end of file condition. If maximum items selected is specified, this request is processed at end of file even if the limit was reached during normal request processing.
E	When input files have reached end of file. Master file, sequentially coordinated files, and transaction files are not available.
S	Only when invoked by another request (S denotes subroutine request).
P	Causes VISION:Builder to generate an IMS SSA. Refer to the *VISION:Builder Reference Guide* for details (P denotes preselection request). [T]
Z	Routes reports according to COLLATE and ROUTE ASL statements. Refer to the *VISION:Builder Reference Guide* for details on the Report Manager and the *VISION:Builder ASL Reference Guide* for details on the COLLATE and ROUTE statements.
0	Before any file processing has begun. [4]
1	After transaction record is read. [4]
2	After master record and transaction record are aligned. [4]

Entry	Points at Which the Request is Executed
M	After master record update and first round coordination are completed.❹
3	After type M. If no type M exists, after updating the master file and first round coordination are completed.❹
4	At any point before types 3 and M. This is used for rejected transactions only.❹
5	Before a Master file record is read.❹

- Request type P cannot be used for graphics.
- Request type Z must be followed by a ##PROC statement, an unlimited number of COLLATE and ROUTE statements, and a closing ##PEND statement (where # is the system delimiter).
- Request type Z should be placed after all of the procedure, report, request, and subfile specifications.

Set name (57-64)

Identifies members of a subroutine request set. See general Rule A. The set name must be the same for each member to be included in the request set.

Leave blank for preselection requests.

Back branch control (65-69)

Can apply to any request type except preselection. This entry limits the number of back branches that can be made from a request during each pass through the request chain or for each pass through all transaction requests of the same type. Back branches exceeding this number are ignored.

Entry	Result
Blank or 0	No back branching is allowed.
1–99999	Mainline request - maximum number of back branches that can be made during one pass through the request chain. Subroutine request - maximum number of back branches that can occur each time the request or set of requests is invoked with a GO/GS/NS statement.

Note: Successful executions of RD operations automatically increment this by 1.

Reinitialize temporary fields? (70)

Specifies automatic resetting to initial values for all temporary fields explicitly defined in this request.

Leave blank for preselection requests.

Entry	Result
Blank or N	No reinitialization is done.
Y	Automatic reinitialization is done for every pass through the request. Only explicitly defined temporary fields are reinitialized to the given initial value or to blanks for character fields or to zero for numeric fields.

Abbreviated format (71)

Controls whether all or a subset of data is printed. See general Rule J.

Leave blank for preselection requests.

Entry	Result
Blank or N	Provides full display of all data elements.
R	Data is displayed in abbreviated format. Repeated selections of the same data field print once.

FD Statement - File Definition

The file definition statement (FD statement) establishes the name of the file definition and describes the physical characteristics of the file such as record format, record or buffer size, records per block.

Note: The aggregate of the FD statement and all LM, LS, Ln, and LR statements defining a file is called a file definition.

Position	Entry
01-08	File name
09-10	Statement type
11-18	File identification
19	Delete?
20	Glossary
21	Record format
22-25	Record Size
26-28	Records per block
29-33	Buffer size
34	GDBI mapped file
35-42	GDBI mapping initialization
43-50	GDBI mapping termination
51-55	GDBI mapped record size

File name (01-08)

Required entry

Name of the file defined to VISION:Builder. See general Rule A.

Statement type (09-10)

Required entry

Identifies a file definition header statement. Enter FD.

File identification (11-18)

For IMS files, provides the DBD (Data Base Definition) name corresponding to the file.

For a mapped file, enter any value to be used by VISION:Builder to initialize the FILEID flag.

For all other files, leave this entry blank.

Delete? (19)

Allows you to delete an entire file definition from the library.

Entry	Result
Blank or N	Definition is added to the library.
Y	Definition is deleted from the library. This entry cannot be specified for an instream definition.
R	Definition is replaced if it is already cataloged; otherwise, it is added to the library. If specified for an instream definition, the instream file definition is used; the file definition from the library is ignored.

Glossary (20)

Selects a glossary listing. See Figure 2-3 through Figure 2-6 at the end of this section for some glossary examples.

Entry	Resulting Glossary
Blank or A	Format 1 alphabetic listing by field name within each segment.
B	Format 1 sequential listing by location of fields within each segment.
1	Format 1 abbreviated listing is alphabetic by field name within each segment.
N	No glossary.
D	Used with the GDBI option to list by segment the requests associated with each segment and command. The virtual segments are listed, also.
X	Format 2 sequential listing by location of fields.
Y	Format 2 alphabetic listing by field name.

Record format (21)

Defines the physical format of the record and the access method to the operating system. Enter one of the following codes:

Operating System Record Format	Possible Structure
F (Fixed Length)	Fixed only
V (Variable Length)	Fixed or Variable
U (Undefined)	Fixed or Variable

Operating System Record Format	Possible Structure
I (Indexed Sequential)	Fixed only
J (Variable ISAM)	Fixed or Variable
K (Key sequenced VSAM)	Fixed or Variable (see Alternate Index Path on page 2-70)
E (Entry sequenced VSAM)	Fixed or Variable (see Alternate Index Path on page 2-70)
A (Alternate Index VSAM)	Fixed or Variable
P (Packed)	Fixed or Variable (see Compressed Records on page 2-70)
D (IMS)	Hierarchical T
H (IMS HDAM)	Hierarchical T
S (DB2)	Relational table D

For DL/I files, a source statement retrieval of an HDAM file definition will have an H in the file definition record format. T

For a mapped file, positions [21–33] must be blank. G

Alternate Index Path

To use an alternate index path to access a VSAM entry sequenced or key sequenced data set, enter K in record format. The JCL must point to the appropriate ESDS or KSDS alternate index path file.

Compressed Records

When a file is defined as packed, the records are compressed before they are stored on a data device and expanded when accessed. The compression is performed without regard to field boundaries or field type and depends only on the data, not the file definition. Compressed records are obtained by deleting certain repeating characters and replacing them with a special 1-byte control field. A string of repeating characters can be compressed when it consists of three or more identical characters. The following character types are recognized.

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

P-type records interface with the operating system as variable length records and buffer size or record size is defined in the same manner as V-type records. As with variable length records, 12 is the minimum record size allowed for packed records. Records with a fixed length logical structure can be defined as packed.

Record size (22-25)

Specifies the number of data bytes in the data portion of a record.

Record Format	Record Size Entry
U, K, E, D, H, A, S	Blank.
F, I	1–9999. If left blank, you must enter the block size.
Unblocked V or P	Maximum record size (LRECL minus 4).
Blocked V, P, or J	Maximum record size (LRECL minus 4). If blank, the maximum record size is computed using the buffer size specification.

Records per block (26-28)

Specifies the number of records in each block. Enter one of the following:

Record Format	Records per Block Entry
V, J, U, K, E, P, D, H, A, S, F (Unblocked), I (Unblocked)	Blank
F (Blocked), I (Blocked)	1–999

Buffer size (29-33)

Specifies the amount of storage required for the buffer, the block size, or the logical record.

- Enter all values right-aligned.
- Leading zeroes are not required.
- 1–32,760 or nnnnK for multiples of 1024 bytes.

Record Format	Buffer Size Entry
F, I	Enter the block size.
V, U, J (Blocked), P (Blocked)	Maximum buffer size (BLKSIZE minus 8). If variable block spanned with a record size greater than 32,760, code maximum record size as xxK.
K, E	Maximum record size (according to the VSAM define cluster specification).

Record Format	Buffer Size Entry
A	Alternate index control interval size.
S	Number of bytes of buffer space VISION:Builder should reserve for the logical record. D
D, H	Enter the maximum amount of main storage required for an entire logical record. I

For memory optimized (MOSAIC) DL/I or relational MOSAIC, enter the size of the largest chain of concatenated segments.

If the operating system supports system-determined block size and the file meets the criteria for this capability, this value is ignored when VISION:Builder processes the file, allowing the operating system to determine the optimum block size and allocate the buffers.

GDBI mapped file (34)

Positions [34-55] are used with GDBI only. Enter M if this is a mapped file.

GDBI mapping initialization (35-42)

Name of the initialization mapping request. See general Rule A.

GDBI mapping termination (43-50)

Name of the termination mapping request. See general Rule A.

GDBI mapped record size (51-55)

Specifies the amount of storage required for the mapped record. Must be at least as large as the root segment. Enter 1–32,760 or nnnnK right-aligned.

```
JAN 15, 1995  12.33.05                                                                    PAGE 4

                                   DETAILED GLOSSARY   BY NAME FOR                  GLOSSARY PAGE 1
                                       FILE DEFINITION - TRANSAC

DATE CATALOGED - 01/15/95  12.15.05                        DATE LAST UPDATED - 01/15/95 12.20.05
CATALOGED BY - MARY SMITH                                  LAST UPDATED BY - MARY SMITH
EXPIRATION DATE - 01/15/99                                 DATE LAST USED - 01/15/95 12.05.05

FILE IDENTIFICATION =                                      RECORD FORMAT = FIXED BLOCKED
NUMBER OF SEGMENTS IN FILE = 1                             RECORD SIZE = 91
NUMBER OF FIELDS IN FILE = 5                               RECORDS PER BLOCK = 1

                              ****************************************
                              *          SEGMENT 1, LEVEL 1          *
                              ****************************************

SEGMENT OCCURS N TIMES = 1                          KEY FIELD 1 = TEMP     TYPE = C    LENGTH = 4
SEGMENT SIZE = 25
NUMBER OF FIELDS IN SEGMENT = 5

-------------------------------------------------------------------------------------------------------------
FIELD   FIELD  FIELD    FIELD   FIELD    DEC   CNT  FIELD EDIT CODES    EDIT     OUTPUT  LINE      REPORT FIELD
NAME    TYPE  LOCATION  LENGTH  RNDING  PLACES FOR  SGMT  ( ) ( ) ( )   LENGTH    WIDTH   NO   ***    HEADING  ***
-------------------------------------------------------------------------------------------------------------
ITEMNO     C     5        7                                              7        6

                                                                                    1    ***   ITEM #   ***
                                                                                    2    ***   NUMBER   ***

ITMCOST    Z    20        6              2                               9        4
                                                                                    1    ***    ITEM    ***
                                                                                    2    ***    COST    ***
REORDPT    Z    16        4                                              6        5
                                                                                    1    ***   REORDER  ***
                                                                                    2    ***    POINT   ***
REORDQTY   Z    12        4                                              6        8
                                                                                    1    ***   REORDER  ***
                                                                                    2    ***   QUANTITY ***
TEMP       C     1        4                                              4        4
                                                                                    1    *** TRANSACTION ***
                                                                                    2    ***    CODE    ***
```

Figure 2-3 File Definition Glossary - Type A or Blank

```
JAN 15 1995  12.33.05                                                                    PAGE 4

                                    DETAILED GLOSSARY BY LOCATION FOR                    GLOSSARY PAGE 1
                                        FILE DEFINITION - TRANSAC

DATE CATALOGED - 01/15/95  12.15.05                      DATE LAST UPDATED - 01/15/95 12.20.05
CATALOGED BY - MARY SMITH                                LAST UPDATED BY - MARY SMITH
EXPIRATION DATE - 01/15/99                               DATE LAST USED - 01/15/95 12.05.05

FILE IDENTIFICATION =                                    RECORD FORMAT = FIXED BLOCKED
NUMBER OF SEGMENTS IN FILE = 1                           RECORD SIZE = 91
NUMBER OF FIELDS IN FILE = 5                             RECORDS PER BLOCK = 1

                              ******************************************
                              *          SEGMENT 1, LEVEL 1            *
                              ******************************************

SEGMENT OCCURS N TIMES = 1                          KEY FIELD 1 = TEMP     TYPE = C    LENGTH = 4
SEGMENT SIZE = 25
NUMBER OF FIELDS IN SEGMENT = 5

-------------------------------------------------------------------------------------------------------------------
FIELD   FIELD  FIELD     FIELD   FIELD    DEC   CNT  FIELD EDIT CODES    EDIT     OUTPUT  LINE      REPORT FIELD
NAME    TYPE  LOCATION  LENGTH  RNDING  PLACES FOR  SGMT  ( ) ( ) ( )   LENGTH   WIDTH   NO   ***    HEADING  ***
-------------------------------------------------------------------------------------------------------------------
TEMP      C     1        4                                               4        4
                                                                                      1    *** TRANSACTION ***
                                                                                      2    ***     CODE    ***

ITEMNO    C     5        7                                               7        6
                                                                                      1    ***   ITEM  #   ***
                                                                                      2    ***   NUMBER    ***

REORDQTY  Z    12        4                                               6        8
                                                                                      1    ***   REORDER   ***
                                                                                      2    ***   QUANTITY  ***

REORDPT   Z    16        4                                               6        5
                                                                                      1    ***   REORDER   ***
                                                                                      2    ***    POINT    ***

ITMCOST   Z    20        6              2                                9        4
                                                                                      1    ***     ITEM    ***
                                                                                      2    ***     COST    ***
```

Figure 2-4 File Definition Glossary - Type B

```
JAN 15 1995  12.33.05

                                    ABBREVIATED GLOSSARY BY NAME FOR                     GLOSSARY PAGE 1
                                        FILE DEFINITION - TRANSAC

                              ******************************************
                              *          SEGMENT 1, LEVEL 1            *
                              ******************************************

KEY1 = TEMP

-------------------------------------------------------------------------------------------------------------------
FIELD              FIELD                EDIT CODES                 OUTPUT                COLUMN
NAME               TYPE                 ( ) ( ) ( )                WIDTH                 HEADING
-------------------------------------------------------------------------------------------------------------------
ITEMNO               C                                               7
                                                                                      ***   ITEM  #   ***
                                                                                      ***   NUMBER    ***

ITMCOST              Z                                               9
                                                                                      ***     ITEM    ***
                                                                                      ***     COST    ***
REORDPT              Z                                               6
                                                                                      ***   REORDER   ***
                                                                                      ***    POINT    ***
REORDQTY             Z                                               6
                                                                                      ***   REORDER   ***
                                                                                      ***   QUANTITY  ***
TEMP                 C                                               4
                                                                                      *** TRANSACTION ***
                                                                                      ***     CODE    ***
```

Figure 2-5 File Definition Glossary - Type 1

```
JAN 15 1995  12.33.05

                                DETAILED GLOSSARY BY SEGMENT FOR                    GLOSSARY PAGE 4
                                    FILE DEFINITION - TRANSAC

                              ****************************************
                              *          SEGMENT COMMANDS AND        *
                              *          MAPPING REQUESTS            *
                              ****************************************


-------------------------------------------------------------------------------------------------------------------
---------- SEGMENT ----------
NAME        NUMBER      LEVEL      COMMAND      REQUEST
-------------------------------------------------------------------------------------------------------------------
CUSTOMER       1          1       GETFIRST     CUSTREQ

.
.
.

INSTALL        60         2       GETNEXT      INSTREQ


JAN 15 1995  12.33.05                                                                          PAGE 5

                                DETAILED GLOSSARY   BY NAME FOR             GLOSSARY PAGE 2
                                    FILE DEFINITION - GDBI0141



                              ****************************************
                              *          SEGMENT 1, LEVEL V          *
                              *          SEGMENT NAME = VSEG1        *
                              ****************************************


COUNT FIELD FOR SEGMENT = ** NONE **                                  VIRTUAL SEGMENT
NUMBER OF SEGMENTS IN FILE = 1
NUMBER OF FIELDS IN SEGMENT 2

-------------------------------------------------------------------------------------------------------------------
FIELD   FIELD  FIELD    FIELD   FIELD    DEC   CNT  FIELD EDIT CODES    EDIT     OUTPUT  LINE      REPORT FIELD
NAME    TYPE  LOCATION  LENGTH  RNDING  PLACES FOR  SGMT  ( ) ( ) ( )  LENGTH    WIDTH    NO   ***    HEADING  ***
-------------------------------------------------------------------------------------------------------------------
VSEG1A     C     1        4                                              4        15
                                                                                          1 ***V SEG 1 FIELD A ***

VSEG1B     C     7        2                                              4        15
                                                                                          1 ***V SEG 1 FIELD B ***
```

Figure 2-6 File Definition Glossary - Type D

Fn Statement - Title/Preface/Formatted Reporting

The formatted reporting statement (Fn statement) allows you to control page/report formatting. With the Fn statement(s), you can print the title anywhere on the page, print field contents in title lines, specify vertical placement of detail lines, and position summary data below the last detail line.

In addition, with the formatted sectional reporting statement (Fn), you can take a more customized design approach to formatting reports by defining the data in each section of the report (that is, override VISION:Builder default column headings by specifying a COLUMN-HEADING SECTION and redefining the column headings for that report).

Position	Entry
01-08	Request Name
09-10	Statement type
11-13	Sequence number
14-72	Text

Request name (01-08)

Required entry

Identifies the request to which this statement belongs. See general Rule A.

Enter the same request name for all statements in the same request.

Statement type (09-10)

Required entry

Identifies a formatted reporting statement. Enter an F followed by a number (n) from 1–9. The number must be the same on all En, Kn, Rn, Pn, and Tn statements in a graph report statement group.

Sequence number (11-13)

Specifies the order of lines on a formatted report. See general Rule I.

Entry	Result
Blank	Assumes order of statement input.
Up to 3 alpha, numeric, or alphanumeric characters	Order in which lines appear on the report. Sequencing is in ascending order.

Text (14-72)

Supplies the specifications for a page of output (including titles, maximum number of detail lines, summaries for all required levels, spacing, and the total number of output lines). Format your page using the following specifications. (Additional information on formatted reporting is in the *VISION:Builder Reference Guide*.)

The system delimiter

(# is the convention chosen for this guide.)

Single delimiter (#)

- Surrounds literals. A literal is data in the form of a character string, numerals, or symbols that prints as it appears on the statement. The delimiter itself is not printed. The maximum length of a literal is 132 characters.
- Terminates field and flag names in the text.

Double delimiter (##)

- Indicates the end of text for an output line. This lets you use more than one line of text to describe a single line of output. The double delimiter appears only at the end of the last line of text and is separated from the text by at least one blank.
- May terminate the nP, nDc, or nS elements.

Section identifier

Defines the start of formatted reporting specifications for a section and defines this as a formatted sectional report.

PAGE-TITLE section.	Specify PAGE-TITLE SECTION
COLUMN-HEADING section.	Specify COLUMN-HEADING SECTION
SUMMARY section.	Specify SUMMARY SECTION

Note: A double delimiter can follow a section identifier.

n#LIT#

"n" is a positive number from 1–page width (maximum 132) indicating the number of times the literal is to be repeated.

- "n" times the number of characters in the literal must not exceed the page width.
- "LIT" is any literal (a literal is data that prints on the report page exactly as it appears in an Fn statement).
- If "n" is not specified, one is assumed. For example, 132#-# would produce a typical VISION:Builder column heading border of dashes.

Number of blanks (nB)

"n" is a positive number from 1–page width that specifies the number of blank spaces before or between individual entries on print lines. Blanks are not specified at the end of a line because each line is automatically filled with trailing blanks according to the page width.

Columnar print position (nC)

"n" is a positive number from 1–page width used to indicate a particular print position on the report page. For example, 20B #NAME# 75C #PAGE# would print the literal NAME starting at position 21 and PAGE starting at position 75.

Number of lines of columnar detail data (nDc##) - formatted report only

- "n" is a positive number from 1–page height specifying the number of detail lines (D) to appear on a report page.
- "C" is the control break level, 1–9, or G, that controls the printing of further detail lines (if this item is omitted, 9 is assumed); the control break level specified must also appear on an Rn statement.
- Vertical spacing is specified on the En statement.
- This entry must appear only once per formatted report, on a line of text by itself. It is also the only reference on the specification to detail line output.
- The placement of this entry in relation to the other lines of text determines the vertical positioning of the detail information on the report with respect to header and trailer groups. In other words, the nDc element also separates the header and trailer groups (signifies the end of one and/or the start of the other) and is required for formatted reports.
- Formatting of detail lines is controlled by the En and Rn statements.
- This entry can be omitted whenever summary report only is requested on the En statement. It is not valid if section identifiers have been specified.
- ## is optional.

Page break level (nP##) - formatted report only

- "n" is a positive number from 0–9 that specifies the control break level at which a page (P) break occurs when there are no detail lines.
- 0P means the page break occurs only when the page has been filled (no space left to print another full set of summaries).
- When a control break occurs at level "n" or at a higher (numerically lower) level, a page break occurs after the trailer group is printed for that level.
- This specification divides the formatted reporting specification into a header and trailer group. The header group prints at the top of each page; the trailer group can be printed several times per page depending on the value of n and the control breaks.
- The nP specification is used only on summary only reports and must appear on a line by itself.
- This element is not valid if section identifiers were specified.
- ## is optional.

Blank lines (nS##)

- "n" is a positive number from 1–page height (maximum 132) that specifies a number of skipped (blank) lines on the reports before, between, or after print lines.
- This entry appears on a line by itself or in conjunction with a line control indicator. It can appear anywhere on the formatted reporting specification as many times as needed.
- When nS is specified on a line controlled by the nA line control indicator, n blank lines (as indicated by the nA element) print if the current control break level number is equal to or higher (numerically lower) than the nA value. For example, a specification of nA 1s yields one blank line at control break level n and higher (numerically lower) and zero blank lines at all other times.
- When nS is specified on a line controlled by the nL line control indicator, one line for the nL plus n blank lines (as indicated by the nS element) print if the current control break level number is equal to or higher (numerically lower) than the nL value. For example, a specification of nL 1s yields two blank lines at control break level n and higher (numerically lower) and one blank line at all other times.
- ## is optional.

Q.FIELDNAME#

- Including field names in text lines causes the contents of the fields to be printed in the positions specified.
- A qualifier (Q) must be followed by a period (.) and the field name. Field names can have up to 8 characters.
- Blanks imbedded in field names are part of the name and must be specified; however, trailing blanks need not be entered. Place a single system delimiter after the last nonblank character.
- All qualifier/field names (Q.FIELDNAME) appearing in a formatted reporting statement must also appear in an Rn statement. If they are used in text lines but are not part of the report detail data, they should be specified as nonprint on the Rn statement (Y in position [26]).
- The qualifiers include the following:

Entry	Result
Blank or N	New master file field name.
O or 0	Old master file field name.
1–9	Coordinated files (1–9) field name.
T	Temporary field name.
X	Transaction file field name.❹
W	Working storage field name.
V	Linkage section field name.
A, B, E, G, H, J, K, M, Q	Array field name.

Flag names

All flag names appearing on the formatted reporting specification (except for PAGE, TODAY, TODAYX, DATE, ISDATE, JULANX, and JULIAN) must also appear on an Rn statement and should be specified as nonprint if they are not part of the detail data.

Control break levels (nL)

- "n" is a positive number from 0–9 or the letter G, specifying the control break level (L) at which fields, flags, and literals (such as all items except summaries) are to print in the summary section or the trailer group.
- This entry is optional and can appear as many times as necessary on any line in the summary section or trailer group.
- When a control break occurs at a level equal to or higher (numerically lower) than n, all information following the nL on the same output line is printed. This level applies until a subsequent nL or a summary keyword appears or until the output line is terminated.
- The termination of text for an output line automatically resets n to 0.
- An n entry of 0 denotes that printing occurs whether or not there is a control break at any level.

Note: This element affects only the contents of a print line. The nA element (below) determines the level for the line to print.

Line control (nA)

- "n" is a positive number from 1–9 or the letter G, specifying the level at which a summary line is to be output.
- This entry is optional and, if used, must be the first entry for a given output line. It can appear only once on each line in the summary section.
- When a control break occurs at a level equal to or higher (numerically lower) than n (or G), the line is processed according to the remaining specifications on that line; otherwise, the summary line is bypassed.

Note: The primary difference between the nA and nL elements is that when a control break occurs, the nL causes blank summary lines to be reserved for more major control break lines and nA does not.

Keyword Q.FIELDNAME#n

- Keyword Q.FIELDNAME specifications are valid in the summary sections only.
- Summaries are indicated by preceding the field description with an appropriate summary keyword and following the field description with a delimiter and the desired level ("n"=1–9 or the letter G).
- When the appropriate summary break occurs (as defined to the Rn statement), the contents of the computed value of the summary is output to the position specified on the summary line.
- Valid keywords are:

Keyword	Summary Type
TOTAL (or TOT)	TOTAL
CUM	CUMULATIVE
COUNT (or CNT)	COUNT
MAX	MAXIMUM
MIN	MINIMUM
AVG	AVERAGE
PCT	PERCENT
RATIO (or RTO)	RATIO

Summary temp fields (STEMPnnd)

STEMPnnd is a temporary field used only in the summary section. STEMP fields hold results of an arithmetic computation to be output in the summary or used in a subsequent expression as an operand A or operand B field. They are defined by VISION:Builder in the order they are referenced.

- "nn" is a number from 01–99 that identifies a particular STEMP field (you can have up to 99 STEMP fields per report).
- "d" is a positive number from 0– 9 denoting the number of decimal places held in the result.
- If left blank, the number of decimal places in the operand A field is transferred to the result field.
- If used in an operand A or B STEMP field, it is illegal.

STEMPnn (pattern)

Output edit patterns can be specified if the STEMP field is to be output.

- See the Rn statement for allowable override edit patterns.
- The length of the pattern specified here can be up to 30 characters.
- If no pattern is specified on STEMP fields, they will be edited according to the output edit attributes last received when used as a result field.

COMPUTE

COMPUTE identifies an arithmetic computation in the summary section. An example of the general format of the COMPUTE statement follows:

```
nA COMPUTE ROUNDED RESULT = OPERAND + OPERAND
```

- nA is the line control indicator.
- COMPUTE is the keyword specifying an arithmetic computation.
- ROUNDED is optional and specifies the type of result wanted.
- RESULT is the STEMP field to hold the result (STEMP01–STEMP99 with or without decimal places).
- = is a required explicit element of a COMPUTE statement.
- +, -, *, or / is the type of arithmetic calculation.
- OPERAND is the A and B operand fields you request to compute the result. They can be from any previously used STEMP fields or any summary specification.

You can also use a numeric value as operand A and/or B. For example, in the statement COMPUTE STEMP012 = TOTAL N.FIELDA * 1.25, 1.25 is a value.

IT Statement - Run Control Group

The item tracking statement (IT statement) is used to maintain and report creation dates/times, expiration dates/times, and other information that pertains to items cataloged in the common library. IT statements can be included as part of run control or in a definition run as part of the definition input group.

Position	Entry
01-08	Run name
09-10	Statement type
11	Generic item name?
12-13	Type code
14-21	Item name
22-29	Master file name ❹
30-35	Expiration date/retention period
36-55	Update ID

- Fully qualified item names include the item type code, item name, and, if the item is a transaction definition, ❹ the name of the master file definition it is associated with.
- If multiple IT statements have generic names, the IT statement that has the greatest number of characters that match the actual item name will be used.
- The IT statement must appear before the item it tracks and after any CT statements in a processing run.
- The IT statement should precede the definition to which it is related. If an IT statement is placed within the definition, the item change will occur when the IT is encountered.

Run name (01-08)

Optional field; all entries are ignored.

Statement type (09-10)

Required entry

Identifies an item tracking statement. Enter IT.

Generic item name? (11)

Indicates whether item name will be a generic name or the full name.

Entry	Result
Blank or N	The full item name is in the item name entry [14-21].
Y	The entry in item name is a generic name.

Type code (12-13)

Enter one of the following codes that identifies the item type.

Code	Definition
Blank	All item types associated with the item name entry.
AD	Array definition.
FD	File definition.
RQ	Request.
RG	Request group.
TB	Table definition.
TD	Transaction group definition.❹

Item name (14-21)

Required entry

If Y is entered in position [11], enter a generic item name. If blank or N is entered in position [11], enter the full name of item. If a generic name is used, you must enter the leading characters of an item's name but less than the full name.

Master file name❹ (22-29)

Identifies the full or generic master file name that is associated with the transaction group definition in the type code entry [12-13].

Entry	Result
Blank	Search for any transaction definitions that are associated with the item name entry [14-21].
Master file name	Search for any transaction definitions that pertain to this master file.

Expiration date/retention period (30-35)

Specifies the actual expiration date or the number of days you want this item to be retained in your library. No action is taken by VISION:Builder when the expiration date is reached; purging of expired items is optional during the MARKUTIL Restore function.

Entry	Result
nnnnn+	Enter 1–5 numerics, right-aligned, followed by a plus sign. This is the number of days you want the item to be retained. The number is converted to an expiration date in the M4LIB.
MMDDYY	Specifies the date (TODAY flag format unless changed in M4PARAMS) you want the item to expire. Where: MM = month, DD = day, and YY = year

Expiration dates in the twenty-first century (20nn) will be computed when the nnnnn+ retention period specification is added to the current date and the result is greater than 1999.

When the expiration date is specified, the century will be the current century if the expiration date is greater than or equal to the current date. Otherwise, the century of the expiration will be the next century.

Updater ID (36-55)

Enter from 1–20 alphanumeric characters to identify the person responsible for cataloging or updating an item.

Note: If the MARKLIBP ITEMTRAK flag is set to 1 or 3 and there is no entry for updater ID, a type 3 message is issued and the item is not cataloged.

Jn Statement - Graphics

See Jn Statement - Graphics on page 5-15 for the Jn statement specifications for graphics output.

Kn Statement - Graphics

See Kn Statement - Graphics on page 5-19 for the Kn statement specifications for graphics output.

LA Statement - File Definition

The automatic table lookup statement (LA statement) identifies the type, input argument, and table name for automatic table lookup. LA statements are used with L0 and L1–L9 statements.

Position	Entry
01-08	File name
09-10	Statement type
11-18	Field name
19	Delete?
30	Field search type
44-51	Table name
52-59	Input argument name

File name (01-08)

Required entry

Name assigned to the file by the file definition. Enter the same name as entered on the FD statement.

Statement type (09-10)

Required entry

Identifies a file definition statement for automatic table lookup. Enter LA.

Field name (11-18)

Required entry

Names the table lookup result field. See general Rule A. The field name must also appear on L0 and L1–L9 statements referencing the table result field.

Delete? (19)

Deletes an LA table lookup statement.

Entry	Result
Blank or N	Table name and attributes are added to the library.
Y	Table name and attributes are deleted from the library.

Field/search type (30)

Indicates the type of table lookup being performed against an argument.

Code	Meaning
Blank or E	Return results of equal argument value only.
N	Return results of equal or nearest argument value.
S	Return results of equal or smaller argument value.
B	Return results of equal or bigger argument value.
I	Return results of linear interpolated argument value (binary tables only).

On a TB statement, table lookup type must be binary for any entry besides blank or E.

On a TB statement, the argument and result data types must be numeric.

Table name (44-51)

Required entry

Identifies the table for automatic table lookup. See general Rule A. This table name must exist in the same library as this file definition.

Note: Table lookup result field characteristics are determined by the table definition TB statement. A field in the result entry assumes the characteristics of the result. If table result field characteristics are modified, the file definition must also be modified by deleting and recataloging the LA statement.

Input argument name (52-59)

Required entry

The name of the field in this file definition that is used as an input argument to the table. See general Rule A. This field name must already exist in the file definition. The input argument field contains the value that is searched for during table lookup.

LB Statement - Run Data Group

The LB statement allows you to assign alias names to segments, fields, or ddnames for files within a run data group. It must immediately follow the RF statement(s).

When you specify memory optimized (MOSAIC) processing, you cannot specify aliases for any segment name or key field name of any segment in the data group.

Position	Entry
01-08	Data group name
09-10	Statement type
11-18	Field or segment name in file
19	Delete?
20-27	Logical file name
28-35	Alias Name
36	Alias Type
37-42	Logical system number

Data group name (01-08)

Required entry

This is the name of the data group to which this file belongs. The name you enter must appear on the previous DB statement.

Statement type (09-10)

Required entry

This entry identifies this as the LB statement. Enter LB.

Field or segment name in file (11-18)

The name of the field or segment for which the alias is provided. This name is the same as the field or segment name in the L0 or LS statement of the file definition for this file. This entry is used only for field or segment name aliases; leave blank for ddname overrides.

Delete? (19)

Entry	Result
Blank or N	Adds this LB statement to the data group. Similar items cannot already exist in the data group.
Y	Deletes this LB statement from the data group.

Logical file name (20-27)

Required entry

Enter the same name (for example, M4OLD, M4CONDn) you entered on the RF statement.

Alias name (28-35)

Required entry

Enter the alias name or override DTF/ddname. See general Rule A. This entry must be provided if you are adding a new alias to the data group or using a VSAM file. Duplicate field and segment names must be avoided and can be alleviated by using alias names. For non-DL/I files, the new ddname, if specified, is searched for in the processing JCL.

Alias type (36)

Required entry

Identifies the item to be changed.

Entry	Result
Blank or F	A field name is changed.
S	A segment name is changed.
D	A DTF/ddname is changed.

Logical system number (37-42)

Valid for VSE only

Allows you to override, at execution time, the VSE fixed set of logical unit assignments specified in MARKUNIT. Enter SYSnnn, where nnn=001–240, PCH, RDR, LST, or IPT.

LM Statement - File Definition

The LM statement defines the relationship between a segment-related GDBI I/O command and the mapping request which processes that command.

Position	Entry
01-08	File name
09-10	Statement type
11-18	Segment name
19	Delete?
44-51	Command
52-59	Mapping request name

File name (01-08)

Required entry

Enter the same 8-character name as on the FD statement.

Statement type (09-10)

Required entry

Identifies a statement defining the relationship between a GDBI I/O command and its mapping request. Enter LM.

Segment name (11-18)

Required entry

Names the segment to which this LM statement applies. This name must also appear in a previous valid LS statement for this record. See general Rule A.

Delete? (19)

Deletes an LM statement from the mapped file definition (FD).

Entry	Result
Blank or N	Adds the LM statement to the mapped file definition (FD).
Y	Deletes the existing LM statement from the mapped file definition (FD).

Command (44-51)

Specifies the command or function that VISION:Builder expects the corresponding mapping request to complete. Depending upon the type of processing you plan to perform, VISION:Builder will issue the appropriate GDBI I/O command.

Command	Action
GETFKEY	Get the first segment with a key value equal to or greater than the supplied start search key value.
GETFIRST	Get the first segment within a parent.
GETNEXT	Get the next occurrence of a segment.
GETKEY	Get the segment whose key value matches the supplied key value.
ADD	Insert a segment.
REPLACE	Replace a segment.
DELETE	Delete a segment.
NOCHANGE	Issued for update-in-place when no action was taken against the segment.

Mapping request name (52-59)

Identifies the mapping request that is to be linked with the GDBI I/O command on this LM statement. The field can be left blank, which will result in a NOP (no operation) for an output command and a segment not found for an input command.

Ln Statement - File Definition

The Ln statement associates the column heading text to a field. Each field in a record is defined by an L0 statement and can have from 1–9 lines of column heading with the L1–L9 statements. The Ln statement is not mandatory; however, if Ln statements are used, the field must have a corresponding L0 statement.

Position	Entry
01-08	File name
09-10	Statement type
11-18	Field name
19	Delete?
44-59	Column heading text

File name (01-08)

Required entry

Enter the same 8-character name as on the FD statement.

Statement type (09-10)

Required entry

Identifies a line of column heading text for the L0 statement with the same field name. Enter Ln (where n is a line number from 1–9).

If a line is duplicated, the first line number is accepted; subsequent line(s) are errors and a message is printed accordingly.

Note: Blank lines in the column heading are added by skipping Ln line number(s).

Field name (11-18)

Required entry

Name of the field where column headings are defined. The field name must be reentered for each nonblank line of column heading. See general Rule A.

Delete? (19)

Deletes a line of column heading text.

Entry	Result
Blank or N	Line of column heading text is added to the field.
Y	Line of column heading text is deleted.

Column heading text (44-59)

Defines the headings to be placed over columns of data on a report.

Statement Type Entry	Line of Column Heading Text
L1	First line
L2	Second line
.	.
.	.
.	.
L9	Ninth line

Enter up to 16 characters for each line of text. The text can consist of any character. A delimiter is optional at the end of a line of text.

- Lines are centered over the columns based on the rightmost nonblank character. If that is any character other than the delimiter, it is assumed to be part of the heading.
- If the last character is the system delimiter, it is not counted as part of the column heading. The delimiter is used if trailing blanks are required after the last nonblank character.

LR Statement - Logical Relationship Definition

The logical relationship statement (LR statement) is used to define logical relationships between the different relational tables used as segments in a logical record.

- It explicitly establishes the relationship between the rows of the different relational tables.
- In addition, it can be used to qualify and filter the rows that will be retrieved from the table.

An LR statement must be provided for each nonroot segment in a relational database and is optional for the root segment.

Position	Entry
01-08	File name
09-10	Statement type
11-18	Segment name
19	Delete?
20-22	Sequence number
23	Logic level
24	Connector
27-44	Relational field name
45-46	Operation
48	Qualifier
49-66	Constant or primary key

File name (01-08)

Required entry

Enter the same name as entered on the FD statement.

Statement type (09-10)

Required entry

Identifies the logical relationship definition. Enter LR.

Segment name (11-18)

Required entry

Enter the name of the segment being qualified by the relational conditions.

Delete? (19)

Allows you to delete an entire logical relationship definition.

Entry	Result
Blank or N	Definition is added to the library.
Y	Definition is deleted from the library.

Sequence number (20-22)

Indicates the order of the statements in the definition. Duplicate sequence numbers within the same segment name are not permitted.

Entry	Result
Up to 3 alpha, numeric, or alphanumeric characters	Sequence is in ascending order.

This field is also used for file definition maintenance to insert LR statements and to delete them from the definition.

- To delete an LR statement, the user must provide file name, LR, segment name, delete=Y, and sequence number. Duplicate sequence numbers are not permitted.
- An LR statement can only be changed by first deleting the old statement and replacing it.

Logic level (23)

Number specifies the sequence of processing multiple AND/OR conditions in a logical expression. The number is similar to using parentheses in an algebraic expression. When logic levels are the same or not specified, AND takes precedence so that A and B or C is always interpreted as (A and B) or C.

Entry	Result
Blank	Level of 0 is assumed.
0–9	Number of pairs of parentheses surrounding the condition if expressed algebraically.

Connector (24)

Indicates the relationship between two or more statements within a logical expression. The first line connector must specify a blank connector.

Entry	Result
Blank, 0, or O	OR relationship between statements.
A	AND relationship between statements.

Relational field name (27-44)

Identifies the column in the relational table to be compared to the item in positions [49-66] of this statement. This name is the DB2 column name as defined in the DB2 catalog.

Operation (45-46)

Required entry

Specifies the logical operation to be used in comparing the relational column named in positions [27-44] to the field or value in positions [49-66] of this statement. The operator you specify results in a select (true) or nonselect (false) condition.

- If the complete expression composed of all the LR statements for a segment is met (true), the table row is selected and becomes a segment in the VISION:Builder record.
- If the condition is not met (false), the row is not selected.

Code	Condition to be Met
EQ	Table column equal to the field or value.
NE	Table column not equal to the field or value.
LT	Table column less than the field or value.
GT	Table column greater than the field or value.
GE	Table column greater than or equal to the field or value.
LE	Table column less than or equal to the field or value.
NL	Table column is null. Positions [49-66] must be blank.
NN	Table column is not null. Positions [49-66] must be blank.

Qualifier (48)

Identifies the type of field or value that follows in positions [49-66].

Entry	Type
Blank	Field name.
C	Character string constant.
D	Decimal constant.

Constant or primary key (49-66)

Specifies the VISION:Builder field name or constant to be compared to the relational table column named in positions [27-44]. This entry must correspond to the qualifier in position [48].

Qualifier	Entry
Blank	A VISION:Builder field name that has been defined in a higher level segment that is on the same leg of the hierarchy as the current segment. It cannot be a type V field.
C	Character string data.
D	Decimal data. It can include a leading sign, the digits 0–9, and a decimal point. The decimal point can also be followed by the digits 0–9. No other characters, including grouping characters, are allowed.

In the LR statements, mixed mode arithmetic is not permitted. LR statements must compare character string to character string or numeric to numeric.

LR statements can be provided for the root segment but, in this case, columns [49-66] cannot be a VISION:Builder field name (that is, qualifiers can only be C or D).

LS Statement - File Definition

The LS statement names the segments in a record.

Position	Entry
01-08	File name
09-10	Statement type
11-18	Segment name
19	Delete?
20-21	Segment number
22	Virtual segment G
30	Suppress duplication? D
31	Segment order
36-43	Authorization ID D
44-61	DB2/SQL external name D

- Each segment in an IMS DL/I file must be named and have the same name as in the DBD (Data Base Definition).
- For relational databases, you must identify each table in the logical record as a segment on an LS statement.
- Each segment in mapped files must be named on an LS statement.
- For nondatabase and nonmapped files, if one segment is named, all segments must be named.

If the FS operator is to be used on a lower level segment, the segment must be named using an LS statement. You must have at least one L0 statement for each LS statement.

File name (01-08)

Required entry

Enter the same 8-character name as on the FD statement.

Statement type (09-10)

Required entry

Identifies a segment name definition statement. Enter LS.

Segment name (11-18)

Required entry

Names the segment. For IMS DL/I files, the name must be the same as the segment name specified in the DBD (Data Base Definition) and associated PCB. For relational, mapped, or other files, this is the desired segment name.

Enter, left-aligned, an 8-character alphanumeric segment name.

Whenever a segment name is added or changed, it is your responsibility to ensure compatibility between the file definition and the DBD.

The segment name is used with the FS operator.

Delete? (19)

Deletes a segment name from the file definition. This entry does not affect fields within the segment; it acts only on the name.

Entry	Result
Blank or N	Adds the segment name to the file definition.
Y	Deletes the existing segment name from the file definition.

Segment number (20-21)

Identifies the segment.

Entry	Segment Number Result
Blank	1 (assigned by VISION:Builder).
1–99	1–99.

Note: To identify a segment with a number greater than 99, use VISION:Workbench.

The same number must be entered for all the entries that define fields in a particular segment (L0 statements).

All subordinate segments must have a segment number larger than the controlling segment and smaller than any following segments.

Virtual segment (22)

Identifies virtual segment.

Entry	Result
Blank	Segments that are not virtual.
V	For virtual segments in a mapped file.

Any uniqueness within a table row causes the row to be mapped.

The entry of Y may cause DB2 to consume additional CPU cycles.

Suppress duplication? (30)D

Enables the elimination of duplicate segments (rows) from the logical record of a relational database.

Entry	Result
N	All applicable rows are mapped to the logical record.
Y	Duplicate rows are not mapped to the logical record (that is, duplicate segments are eliminated).

Segment order (31)

Specifies the segment order.

Each segment occurrence within a parent segment must have a unique key field value. Segment order applies to the values of the segment keys after being converted to their field types as defined in the transaction file definition.

Entry	Order
Blank	Unspecified order and unordered segments.
A	Segments are ordered within parent segments by segment key field(s) in ascending sequence.
D	Segments are ordered within parent segment by segment key field(s) in descending sequence. Descending root segments are permitted for relational tables.

- If a segment is keyed in the DL/I file, ascending is the only appropriate segment ordering.I
- The entries of A and D may cause DB2 to consume significant additional CPU cycles. The ordering is accomplished by adding an ORDER BY clause to the SQL SELECT statement for that segment.D

Authorization ID (36-43)

Specifies the authorization ID of the person who created the relational table to be used to build the logical record. If all relational tables in the logical record have the same authorization ID, this entry can be left blank. To obtain the authorization ID from another source, leave this entry blank.

The authorization ID can also be specified on the RF statement. The order of precedence is:

- The entry on the RF statement, if provided.
- The entry on the LS statement, if provided.

If an authorization ID is provided, VISION:Builder supplies DB2 with a prefixed table name in the form of:

```
AUTHORIZATIONID.TABLENAME.
```

If an authorization ID is not provided, VISION:Builder supplies DB2 with an unprefixed table name. In this case, DB2 uses the current SQL authorization ID. This comes from the SQLID parameter provided on the RP statement in the application.

If an RP statement is not provided, DB2 uses the signon user ID as the table prefix. Under OS/390, this is entered on the JOB statement. Under VSE and CMS, this is entered on the RP statement.

DB2/SQL external name (44-61)

Identifies the relational table that corresponds to this segment. This is a required entry for relational databases. This can be the name of a physical relational table or a relational view.

This entry can contain an SQL delimited name.

LX Statement - File Definition

The LX statement is used to specify long field names. Long field names can be up to 30 characters long. It allows for a more descriptive field name or a place to hold the name of a field in a file or database as defined by another language such as COBOL. It cannot be the same name as one on an L0 statement for another field or another LX statement in the same definition. The long name on the LX statement can be the same as the short name on the L0 statement for any individual field.

Position	Entry
01-08	File name
09-10	Statement type
11-18	Base field name
19	Delete?
30-59	Long (alternate) field name

File name (01-08)

Required entry.

The same name as entered in columns 01-08 of the FD and L0 statements.

Statement type (09-10)

Required entry.

Identifies the LX statement. Enter LX.

Base field name (11-18)

Required entry.

Field name as entered on the L0 statement for this field.

Delete? (19)

Deletes individual field definitions.

Entry	Result
Blank or N	Catalogs the field attributes in the file definition.
Y	Deletes the field from the file definition.

Long (alternate) field name (30-59)

Long field name assigned to this field; up to 30 characters long.

L0 Statement - File Definition

The L0 statement defines the attributes of a field (such as size, data type, and output editing). An L0 statement is required for each field defined in a record.

Position	Entry
01-08	File name
09-10	Statement type
11-18	Field Name
19	Delete?
20-21	Segment number
22	Level number
23-26	Field location
27-29	Field length
30	Field/search type
31	Segment key
32	Field rounding?
33	Decimal places

Position	Entry
34-35	Count field for segment number
36-38	This segment occurs n times
39-43	Output edit
39	Floating
39	DBCS data state
40	Filling
41	Trailing
42-43	Length
44-51	DL/I alias (I)
44-61	DB2/SQL external name
44-71	DB2/SQL external name (D)

File name (01-08)

Required entry

Identifies the file containing the field. Enter the 8-character name as on the FD statement.

Statement type (09-10)

Required entry

Identifies a statement defining the attributes of a field in the file definition. Enter L0.

Field name (11-18)

Enter the name of the field defined, the name of the generic key field defined, or the name of the result field if using automatic table lookup. See general Rule A.

Delete? (19)

Deletes individual field definitions.

Entry	Result
Blank or N	Catalogs the field attributes in the file definition.
Y	Deletes the field from the file definition.

Note: D1, D2, LA, LX, and L1–L9 statements corresponding to an L0 marked for deletion are automatically deleted.

Segment number (20-21)

Identifies, by number, the segment to which the field belongs. A virtual key field must be in the root segment.

Entry	Segment Number
Blank	1
1–99	1–99

Note: To identify a segment with a number greater than 99, use VISION:Workbench.

Level number (22)

Indicates the subordination of segments. A virtual key field must be in the root segment.

Entry	Level
Blank	1 (assumed by VISION:Builder).
1–9	1–9 (1 is the root segment; 2 to 9 are lower level segments).
V	V for virtual segment for a mapped file.

Field location (23-26)

Required entry

Indicates the position of the field in a segment.

Entry	Result
1–9999	Location of the field relative to the beginning of its segment.

Note: Leave blank for type V fields, virtual key fields, or automatic table lookup result fields.

Field length (27-29)

Required entry

Specifies the length of a field in characters or bytes.

Field Type	Range
Blank or C	1–255 bytes (characters).
Z	1–15 bytes.
P	1–15 bytes (29 digits plus sign).
V	1–999 or 1–99H (not including 4-byte header), where H implies 00 (for example, 15H = 1500).
L	3 to 8 bytes.
S	2 to 8 bytes.
E	4 bytes only (range10^{-78} to 10^{75}).
F	1–4 bytes with ranges as follows:
D	4 bytes only (interpreted as a Lilian date when displayed).

Field Length	Range
1 byte	-128 to 127
2 bytes	-32,768 to 32,767
3 bytes	-8,388,608 to 8,388,607
4 bytes	-2,147,483,648 to 2,147,483,647

- For a virtual key field, the length is equal to the combined length of all fields specified by the SRCH parameter of the XDFLD statement in the DBD.
- For the primary key field of a multi-column keyed segment, the length is equal to the combined length of all fields that overdefine the primary key field.

Field/search type (30)

Indicates the type of data in the field.

Code	Definition	IMS Field Code
Blank or C	Character string	C
Z	Zoned decimal number	C
P	Packed decimal number	P
F	Fixed point binary number (negative values are stored in 2's complement notation)	X, F, H
E	Floating point number	X
V	Variable length	-
R	Automatic table lookup result field	-
L, S	Time processing	P
D	Date (Lilian only)	X, F

- The IMS field codes are important only when the fields are to be used in SSAs.
- An entry of V is not allowed for a memory optimized (MOSAIC) relational file.
- Enter C for primary key of a multi-column keyed segment.
- Enter C for virtual key field.
- Z field type does not apply in IMS. Nonkey fields can have a type Z format as long as it corresponds to the internal format of the data.

Segment key (31)

Required entry

Identifies a segment key field.

Entry	Result
Blank	Type V fields, nonkey fields, and component fields of the primary key for a multi-column keyed segment.
1–9	1–9 key fields per segment (1 is the highest and 9 is the lowest record key).
S	This nonkey field is a search field and can be used in segment search arguments for preselection and memory optimized (MOSAIC) transaction processing. I
V	This field is a virtual key field. No other key fields, except search fields, can be named for this segment. V is valid for a root segment only. I

- ISAM files (record type I, J) record keys cannot begin in position 1 if records are to be deleted from the files.
- Key sequenced VSAM files (record type K) allow only 1 record key to be defined.
- Numeric record keys can be negative; however, they will be collated ahead of positive values for numeric (types Z, P, F, E) field types.
- Enter the number 1–identify the IMS segment key field.
- If an IMS database is updated, all subordinate segments must be uniquely keyed to IMS.
- S and V are valid only in IMS database definitions.
- Usually, a field that is a key field must also be a mapped field (that is, a mapping name starting in column 44 must be provided), unless you use the multi-column keyed segment capability. In this case, it can be left blank if the key field is defined as a character field. Refer to the section on multi-column keyed segments in the *VISION:Builder Reference Guide* for other requirements and restrictions. D
- If more than one key is defined for the segment, only the primary key is involved in a keyed read operation (start/end search, direct-read, checkpoint/restart, ICF coordinated files, user read coordinated files, MOSAIC find segment [FS] with a key).

Field rounding? (32)

Used only for numeric fields in arithmetic, replacement, table lookup, or transaction[4] updating operations. If this field is used as operand C on a PR statement, the contents are rounded, not truncated.

Entry	Rounding?
Blank or N	No (Result will be truncated)
Y	Yes (Result will be rounded)

Decimal places (33)

Indicates the number of decimal places in a numeric field (types Z, P, or F) or time constants.

Entry	Decimal Places	Field Type
Blank	0 (assumed)	C, E, V
0–9	0–9	Z, P, F, S, L

- In type Z fields, the number of decimal places must not exceed the field length.

  In type P fields, the number of decimal places must not exceed twice the field length minus one.

  In type F fields, the maximum number of decimal places depends on the field length, as shown.

Length (bytes)	Maximum Number of Decimal Places
1	3
2	5
3	7
4	9

- Leave blank for type V fields.

Count field for segment number (34-35)

Required in the parent of variably occurring lower level segments. Specifies the number of a segment type for variably occurring segments (field types must be Z, P, or F). This field is automatically updated. Do not change it. Leave blank when using fixed length or indexed sequential files. This entry is optional for IMS files, mapped GDBI files, and relational files.

Entry	Count Field
Blank	No count field.
2 to 99	Segment number whose occurrences this field counts.

This segment occurs n times (36-38)

Specifies the number of fixed occurrences of a segment type in a record. This specification is made for the primary key of the segment. Leave this entry blank for type V fields, IMS files, mapped GDBI files, or relational files.

Entry	Result
Blank or 0	For the root segment or variably occurring segment types.
1–999	Number of fixed occurrences of this segment type for each occurrence of its parent segment.

Note: If 1–999 is entered in this field, its segment key [31] cannot be blank.

Output edit (39-43)

The output edit entries (floating, filling, trailing) are used only with numeric fields (types P, Z, F). They edit the field before it is printed in a report.

Commas print in all fields when preceded by a significant digit. To suppress the printing of commas, use the edit suppress character or the override "picture" edit on the Rn statement [53-67].

Floating/edit suppress

An entry other than Z in this column "floats" the value of the entry and prints it to the immediate left of the first significant digit in the report.

- If this column is blank, a leading blank is printed if positive, a leading minus sign if negative.
- If this column contains a Z, all commas, leading zeroes to the left of the decimal point, and the decimal point, if specified, are suppressed.

Floating (39)

This editing code applies to numeric field types P, Z, and F in position [30].

This entry prints immediately to the left of the first nonblank character.

Code	Results
Blank	When no output edit codes are specified, a leading blank prints if the value of the field is positive, a leading minus sign if the value of the field is negative; commas and decimal points print if decimal places are specified. A zero value in a field where a decimal place is specified prints as a decimal point followed by as many zeroes as decimal places.
$	A floating dollar sign prints before the first value in a column when no summaries are taken. When summaries are taken, the $ also prints to the left of the summary value.
+	A leading plus sign (+) is printed if the value of the field is positive, a leading minus sign if negative.
-	A leading minus sign (-) is printed if the value of the field is negative; positive values are unsigned.
Z	The printing of commas, decimal points, and leading zeroes to the left of the decimal point is suppressed.
Any other character	A floating lead character prints with the same attributes as $.

Note: Minus signs print, but no space is allocated for the minus signs.

DBCS data state (39)

This editing code applies to character field type C in position [30].

This entry is used to indicate the Double Byte Character Set (DBCS) data state.

Code	Data State
Blank	EBCDIC.
I	DBCS data with shift characters embedded in the data. This data is submitted to the printer without modification.
E	DBCS data that does not contain any embedded shift characters. This data is submitted to the printer with shift characters embedded in the data.

Filling (40)

This editing code applies to numeric field types P, Z, and F in position [30]. This entry prints in every position, from the leftmost to the first nonzero digit.

Code	Result
Any character	Replaces leading zeroes.

Trailing (41)

This entry prints a trailing character.

Code	Results
+	A trailing plus sign (+) is printed if the value of the field is positive, a trailing minus sign if negative.
-	A trailing minus sign (-) is printed if the value of the field is negative; positive values are unsigned.
)	Negative field values are enclosed in parentheses. If no filling character is specified, the left parenthesis prints before the first significant digit or decimal point, whichever comes first. If a floating character and this character are specified, both can print. The floating character prints inside the parentheses ($43.50). Only a single floating sign is permissible with the trailing ")".
C	Prints a trailing "CR" for a negative value. Blanks follow a positive value.
D	Prints a trailing "DB" for a negative value. Blanks follow a positive value.
Any other character	Prints as a trailing character for negative values only.

Length (42-43)

The number of positions required to print the field.

Fixed Length	Entry	Resulting Output
Fixed Length	Blank	VISION:Builder computes the length, including edit characters.
	The number of positions to print the field (including edit characters)	You compute the length.
Type V field	Blank	Length is the shorter of the report page width minus spaces before column or the defined field length.
	Any number up to the maximum field length	The field text folds within this width until it is exhausted.

DL/I alias (44-51)

Enter the 1–8 alphanumeric character name that corresponds to the name specified in the DBD.

This specification is used when defining an IMS database with nonunique segment key/search field names.

This entry, used in the L0 statement that defines a segment or virtual key/search field, provides the nonunique name specified in the DBD. VISION:Builder uses this name only to construct segment search arguments (SSAs).

DB2/SQL external name (44-61) or (44-71)

Enter the 1–18 character name of the relational column that is to be transferred from the relational table into the VISION:Builder logical record. If you do not enter an external DB2 or SQL/DS name, the field will exist in the VISION:Builder logical record, but no transfer of data will take place.

This name will also be used when tables are created. The name can be entered for all record formats, but has relevance only when the field is input from or output to a relational table or when creating a relational table using subfiles.

This entry is required for key fields in relational database definitions, unless you use the multi-column keyed segment capability. In this case, this entry can be left blank if the key field is defined as a character field. Refer to the section on multi-column keyed segments in the *VISION:Builder Reference Guide* for other requirements and restrictions.

DB2 date/time fields can be defined to VISION:Builder as character fields. However, the default initial value of blanks is not a valid date/time format to DB2. The user must ensure that a valid value is entered in the field prior to writing to a file.

The entry can be up to 28 characters long [44-71] and contain SQL names, SQL delimited names, SQL scalar functions, or SQL column functions.

Note: SQL external names are only checked to make sure that, when specified, the first character is not a blank. Actual verification will take place in the decode phase of a processing run when the information is used (that is, the segment or field is referenced).

OC Statement - Run Control Group

The own-code statement (OC statement) is used when own-code hooks are activated and user written code (own-code) has control of predetermined points during the run. See the *Environment Guide* for details of specific hooks.

Position	Entry
01-08	Run name
09-10	Statement type
11-18	Module name
19	Relocatable
20	Interface control
21-28	Identifier

Run name (01-08)

Required entry

Identifies a particular run. Enter the same name entered on the RC statement run name.

Statement type (09-10)

Required entry

Identifies the own-code statement. Enter OC.

Module name (11-18)

Identifies the own-code control module. Enter the name of the module used in the run. If own-code has been installed with static integration, name is ignored.

Relocatable? (19)

VSE users only

Determines whether the routines are dynamically relocatable. This enables VISION:Builder to make optimum use of storage space. This entry is ignored if own-code has been installed with static integration.

Entry	Is Module Relocatable?
Blank	No
Y	Yes

Interface control (20)

Not used by VISION:Builder. Reserved for future use.

Identifier (21-28)

Not used by VISION:Builder. Available as a means of communication with your own-code control module.

Activate own-code exits

Required entry

Exit1	[29-30]	Exit5	[37-38]	Exit9	[45-46]
Exit2	[31-32]	Exit6	[39-40]	Exit10	[47-48]
Exit3	[33-34]	Exit7	[41-42]		
Exit4	[35-36]	Exit8	[43-44]		

Identifies the own-code hooks activated during the run. See general Rule D. Enter 2-digit own-code hook identifiers as required. The first should be in [29-30], the second in [31–32], and so on. A maximum of ten (10) hooks can be specified on one statement; however, you can use as many statements as you need.

PA Statement - Run Control Group

The program analyzer statement (PA statement) is an optional part of the run control group. The PA statement is used to generate listings that can be used for enhanced program documentation and assistance in debugging your VISION:Builder programs. Refer to the *VISION:Builder Reference Guide* for a complete description of the program analyzer option.

Position	Entry
01-08	Run name
09-10	Statement type
11	Conversion messages?
12	Cross reference?
13	Execution trace?
14-18	Maximum lines
19-23	Block size
24	File format

Run name (01-08)

Required entry

Enter the same name as entered on the RC statement. See general Rule A.

Statement type (09-10)

Required entry

Identifies a program analyzer statement. Enter PA.

Conversion messages? (11)

Specifies whether numeric conversion messages will be listed.

Entry	Result
Blank or N	Numeric conversion messages are suppressed.
Y	Numeric conversion messages are printed.

Cross reference? (12)

Specifies whether data will be extracted for the creation of cross-reference reports:

- Field name cross-reference.
- Branch location cross-reference.
- Constant and pattern cross-reference.
- File cross-reference.

Entry	Result
Blank or N	Data will not be extracted for the cross-reference reports.
Y	The M4PAOUT file will be used and will contain data for the cross-reference reports.

Execution trace? (13)

Specifies whether data will be extracted for either of the execution trace reports, one for reporting standard flag settings and the other for GDBI mapping request events.

Entry	Result
Blank or N	Execution trace records will not be generated.
Y	Execution trace records will be output to M4PAOUT.

Maximum lines (14-18)

Specifies the maximum number of lines to be provided for the execution trace reports. This is the total number of lines for both reports.

Entry	Result
Blank	Value defaults to the installation standard (see M4PARAMS PALTRCMX in the *VISION:Builder for OS/390 Installation Guide* or the *VISION:Builder for VSE Installation Guide*).
nnnnn or nnnnK	A number, right-aligned or in the form nnnnK, where K indicates a value of 1,024.

Block size (19-23)

Specifies the physical block size to be used for M4PAOUT. If the operating system supports system-determined block size and the file meets the criteria for this capability, this value is ignored and is determined by the operating system.

Entry	Result
Blank	Value defaults to the installation-specified block size for M4REPO (see M4PARAMS in the *VISION:Builder for OS/390 Installation Guide* or the *VISION:Builder for VSE Installation Guide*).
nnnnn or nnnnK	A number, right-aligned or in the form nnnnK, where K indicates a value of 1,024.

File format (24)

Specifies the physical format for M4PAOUT.

Enter blank or V for variable blocked.

Pn Statement - Report Preface

The preface statement (Pn statement) is used to enter preface information for a report.

If the Pn statement is used in a report that produces a graph, see Chapter 5, *Graphic Fixed Format Statement Listing* for the Pn statement specifications for graphics output.

Position	Entry
01-08	Request name
09-10	Statement type
11-13	Statement sequence
14-72	Text

Request name (01-08)

Required entry

Identifies the request to which this statement belongs. See general Rule A.

Enter the name exactly as it appears on other statements in the request.

Statement type (09-10)

Required entry

Identifies this as a preface statement or a dynamic report line modification statement. VISION:Builder recognizes these entries by the use of the keyword MARKXREP in the text entry [14-72] on the first Pn statement.

Enter a P followed by a number 1–9. The number (n) following the character P must be the same as the number (n) associated with the Rn statement(s).

Statement sequence (11-13)

Indicates the order in which statements are processed. See general Rule I.

Entry	Result
Blank	Order of statement input is assumed.
Up to 3 alpha, numeric, or alphanumeric characters	Order in which lines appear on the report. Sequencing is in ascending order.

If the number of preface lines entered on the Pn statement is less than 12, the preface is automatically centered vertically on the page.

If it exceeds 12, the preface begins at the top of the page unless blank lines were specified.

To specify blank lines, assign sequence numbers on all lines. Enter the text for the preface line in positions [14-17] or a delimiter in position [14].

Text (14-72)

Specifies the text that becomes the preface to the report. All characters are permissible; however, the delimiter ends a line of text. The scan for the delimiter continues until it is found. It is possible that a line of text output on a preface page may consist of multiple Pn statements. If the number of lines exceeds the maximum that can be accommodated on the specified page size (En statement [17-19]), the page will overflow to the following page(s).

Dynamic Report Line Modification

If the first report statement in position 14 is MARKXREP, then the Pn statement can be used to enter dynamic report line modification statements.

The entries you make in the text positions [14-72] specify how to dynamically format a report. The specifications consist of keywords, positional indicators (nnn), operators (=,1), and operands. With the exception of the keyword MARKXREP, the text can appear in any position in the text entry so long as the keyword is first. The delimiter (#) ends a line.

- Because these operations take place after the report line is formatted, generate a sample report first to determine values for positional indicators and operands where needed.
- Dynamic reporting specifications can be input in any order; however, execution is in the order specified by the keywords that follow.
- Comments can be inserted in Pn statements anywhere after the keyword.

MARKXREP

Invokes the dynamic report line modification capability. The keyword MARKXREP must be entered beginning with position [14].

Note: The keyword MARKXREP must be the first Pn statement in a report request; therefore, it is the lowest sequence number.

TITLE LINENO=nn#

Exempts a specified number of lines (nn = any 2-digit number) from processing by other dynamic report line modification statements.

On an output printer, this specification takes effect whenever a skip to new page (channel 1 punch) occurs.

The TITLE LINENNNO=nn# keyword allows you to modify line spacing or perform calculations without affecting title lines and column headings.

SKIP c POSnnn=x#

Invokes ASA carriage control characters on specific report lines. This is useful when using preprinted forms.

c — Carriage control channel to be invoked (2 to 9) or print without spacing (+).

nnn — Any 3-digit number between 001 and 132 that indicates the position where the x character is or is not found.

x — Character (any printing or nonprinting character) to cause the invoking of c to take place, if:

= x character is found.

* x character is not found.

Up to three SKIP statements are allowed per report.

Multiple fonts per printed line are allowed for page printers (laser printers).

NOSPACELINES

Suppresses output of any blank line(s). When lines are suppressed, the line count is decremented; pages are filled as if the blank lines had never been output to the report file. This specification does not affect title lines or those inserted by the SPACEBEFORE or SPACEAFTER keywords.

NOCALC POSnnn=x#

Allows lines meeting specific logical conditions to be exempted from some calculations and formatting (PERCENT, RATIO, ROUND, DECIMALEDIT, SUPPRESS SPACE).

nnn — Any 3-digit number between 001 and 132 to indicate the position where the x character is or is not found.

x — Character (any printing or nonprinting character) to cause the operation of percent ratio, round, decimal edit, or suppress space to be bypassed, if:

= x character is found.

* x character is not found.

The NOCALC keyword can be used to avoid editing or calculating on subtitles or to allow performing some operations on detail lines but not on summaries.

Up to three NOCALC statements are allowed per report.

PERCENT, RATIO, ROUND

Specifies arithmetic operations that allow division or rounding based on data in the output line. These three operations are executed in the processing phase, in order of input, after NOCALC, and before DECIMALEDIT.

PERCENT A=(nnn,pppp) B=(nnn,pppp) C=(nnn,pppp)

Calculates $\frac{A}{B} \times 100 = C$. The picture used for each field must correspond exactly to the output. The operand A and B pictures are determined by the VISION:Builder output; the operand C picture is determined by you.

nnn — Location of first character associated with the field involved in the calculation (001–132).

pppp — Picture edit in the following ranges:

Operand A and B:

- 1–11 digits before the decimal point.
- Up to 4 digits following the decimal point.
- Include any leading or trailing minus signs, as well as commas and decimal points required by the data.

Operand C (Result field):

- 1–11 digits before the decimal point.
- Up to 2 digits following the decimal point.
- Commas are not allowed.
- A leading minus sign always prints for negative values.

- The picture scan ends when the first nonnumeric character is encountered.
- A trailing % is required in the picture.
- Division by 0 or a field that does not agree with its picture results in a blank output field rather than the invalid field indicator (*).
- Result values that do not fit in the edit picture for operand C produce a plus sign (+) on the output.
- Starting locations of two different operands (nnn) can be the same to allow you to replace a standard output field with one that has been calculated.

RATIO A=(nnn,pppp) B=(nnn,pppp) C=(nnn,pppp)

Calculates $\frac{A}{B} = C$. The calculation is simple division and can be used as such or as a true ratio.

nnn Location of first character associated with the calculation (001–132).

pppp Picture edit in the following ranges:

Operand A and B:

- 1–11 digits before the decimal point.
- A maximum of 4 digits following the decimal point.
- Include any leading or trailing minus signs, as well as commas and decimal points required by the data.

Operand C (Result field):

- 1–9 digits before the decimal point.
- At least 1 digit after the decimal point, up to a maximum of 4.
- Commas are not allowed and a leading minus sign always prints for negative values.

In all result fields (operand C) calculated by the dynamic report line modification capability, leading zeroes are suppressed up to the position before the decimal point.

Starting locations of two different operands (nnn) can be the same to allow you to replace a standard output field with one that has been calculated.

ROUND A=(nnn,pppp)

Provides rounding (half adjusting) and formatting.

nnn Location of first character associated with the calculation made on it (001–132).

pppp Picture edit in the following ranges:

Operand A (Result field):

- 1–11 digits before the decimal point.
- Up to a maximum of 4 digits following the decimal point.
- Leading or trailing minus signs can be substituted for one of the digits.

Rounding is done on the field specified in nnn. If the number is positive, 5 is added to the rightmost digit; if negative, 5 is subtracted. In both cases, the last digit of the result is truncated, including the decimal point if there is only one place in the original data field.

Formatting is a result of the picture characteristics of the operand. With this function:

- A sign can be moved from leading to trailing.
- Extra units positions can be inserted.
- A number between +1 and -1 will be shown with a zero before the decimal point.
- A zero value will print as blanks unless it becomes zero only after the rounding function is completed.

Commas are permitted in the picture and appear in the result when printed.

If the result of the rounding would overflow the size of the picture, the maximum possible value is inserted.

DECIMALEDIT

Provides extra formatting features for fields with a decimal point, no leading minus sign, and values between +1 and -1. The results are similar to those described for the ROUND keyword.

When the DECIMALEDIT keyword is used, data values on all output lines not exempted by the TITLE LINENO or NOCALC keywords are modified when they fall in the specified value range:

- On all zero results, printing is suppressed and a blank is inserted in the output.
- At the same time, on all fields with values less than one with at least one place after the decimal point and no leading minus sign, a zero is printed before the decimal point.

These two operations occur simultaneously and cannot be invoked separately. When it is necessary to separate the functions, use the ROUND keyword.

The location of the fields is determined by a scan of the first line on the report not controlled by the TITLE LINENO or NOCALC keywords. This determines the position of all decimal points.

- If a column exists without a decimal point on that first line, DECIMALEDIT will not affect that column on any line.
- Similarly, a period in a character string of the first line will be interpreted as a decimal point and DECIMALEDIT will be attempted at that column on all nonexempt lines.

SUPPRESS SPACE=(nnn,nnn)

Suppresses printing of lines having spaces in specific positions. The parameters nnn,nnn specify the starting and ending positions to be examined. When the print image between and including those positions is blank, the report line is suppressed.

- The internal line count is adjusted taking into account the effect of the ASA carriage control character that can appear in position zero (0) of the line. If the carriage control is not one that causes single-, double-, or triple-spacing (blank, 0, or +), the line is not suppressed. Consequently, the first line at the top of the page is never suppressed.
- Up to three SUPPRESS SPACE statements are allowed for each report.
- The SUPPRESS SPACE action occurs after arithmetic and DECIMALEDIT operations so that line suppression can be coded to appear as the result of a calculation or on lines that have a zero value in a field.

SPACEBEFORE POSnnn[=*]x#
SPACEAFTER POSnnn[=*]x#

Inserts a blank line before and/or after a specified line (nnn).

nnn — Any 3-digit number between 001 and 132 that indicates the position where the x character is or is not found.

x — Character (any printing or nonprinting character) that invokes the operation, if:

= — x character is found.

* — x character is not found.

These keywords override the NOSPACELINES and SUPPRESS SPACE keywords.

- If you try to insert two blank lines by following SPACEAFTER with SPACEBEFORE, only one will result.
- The keyword is effective only on single-spaced reports because it changes carriage control characters instead of creating actual blank lines.
- Up to three occurrences of each keyword are allowed for each report.

LINE SIZE=(nnn,nnn)(,MOVEPAGENO)#

The LINE SIZE=(nnn,nnn) part of this keyword allows you to reduce the size of a page from the size specified in the first nnn to that of the second nnn, where nnn is any position from 001–132. Fields beyond the truncated page size will not appear on the report.

The MOVEPAGENO part of this keyword is optional. When it is used, the page number (valid only for upper right page positions) appears in the same relative position on the narrowed page.

Note: The report title is not automatically recentered by VISION:Builder when the LINE SIZE parameters are specified. To recenter the title on the narrowed page, the Tn statement(s) should contain trailing blanks before the delimiter.

Diagnostic messages

The dynamic report line modification diagnostic messages are listed in the following table:

Message	Explanation
>3 SEPCHAR	EXREP allows a maximum of three separator characters between integers. This has been exceeded.
>13%/RAT.	A maximum of 13 PERCENT and/or RATIO keywords are allowed in one report. This limit has been exceeded.
=/1MISS.	The equal or not equal operator is invalid or missing for the keyword used.
A-DEC.>4	The number of places after the decimal point in the operand A picture of the PERCENT/RATIO keyword must not exceed four.
A-END>PGW	The last position of the operand A of the PERCENT/RATIO keyword is specified outside the print line width.
A-INTEG=0	No digits were found before the decimal point in the operand A picture of the PERCENT/RATIO keyword or the ROUND keyword.
A-INTEG>11	The number of digits before the decimal point in the operand A picture of the PERCENT/RATIO keyword must not exceed 11.
A-OP MISS.	The operand A of the PERCENT/RATIO keyword is missing.
A-PICT.END	The operand A picture of the PERCENT/RATIO keyword is not terminated by a right parenthesis. This message may also occur when the last character of the picture is the decimal point.
AADR>PGW	The positional indicator of the operand A of the PERCENT/RATIO keyword is outside the print line width for the keyword used.
AADR WRONG	The first positional indicator is not a 3-digit number for the keyword used. Leading zeroes must be supplied.
ADDR>PGW	The first positional indicator is outside the print line width.

Message	Explanation
ADDR WRONG	The first positional indicator is not a 3-digit number for the keyword used. Leading zeroes must be supplied.
ADR1>PGW	The first positional indicator of the SUPPRESS SPACE keyword is outside the print line width.
ADR1 WRONG	The first positional indicator of the SUPPRESS SPACE keyword is not a 3-digit number. Leading zeroes must be supplied.
ADR1>ADR2	The first positional indicator is greater than the second in the SUPPRESS SPACE keyword.
ADR2>PGW	The second positional indicator of the SUPPRESS SPACE keyword is outside the print line width.
ADR2 WRONG	The second positional indicator of the SUPPRESS SPACE keyword is not a 3-digit number. Leading zeroes must be supplied.
B-DEC>4	The number of places after the decimal point in the operand B picture of the PERCENT/RATIO keyword must not exceed four.
B-END>PGW	The last position of the operand B of the PERCENT/RATIO keyword is specified outside the print line width.
B-INTEG=0	No digits were found before the decimal point in the operand B of the PERCENT/RATIO keyword.
B-INTEG>11	The number of digits before the decimal point in the operand B picture of the PERCENT/RATIO keyword must not exceed 11.
B-OP MISS.	The operand B of the PERCENT/RATIO keyword is missing.
B-PICT.END	The operand B picture of the PERCENT/RATIO keyword is not terminated by a right parenthesis. This message may also occur when the last character of the picture is the decimal point.
BAD ASACHAR	The ASA character is not between 2 and 9 in the SKIP keyword.
BADR>PGW	The positional indicator of the operand B of the PERCENT/RATIO keyword is outside the print line width.
BADR WRONG	The positional indicator of the operand B of the PERCENT/RATIO keyword is not a 3-digit number. Leading zeroes must be supplied.

Message	Explanation
C-DEC MISS.	No decimal places were found after the decimal point in the operand C picture of the PERCENT/RATIO keyword.
C-DEC>4	The number of places after the decimal point in the operand C picture of the PERCENT/RATIO keyword must not exceed four.
C-END>PGW	The last position of the operand C of the PERCENT/RATIO keyword is specified outside the print line width.
C-INTEG=0	No digits were found before the decimal point in the operand C of the PERCENT/RATIO keyword.
C-INTEG>9	The number of digits before the decimal point in the operand C picture of the RATIO keyword must not exceed nine.
C-INTEG>11	The number of digits before the decimal point in the operand C picture of the PERCENT keyword must not exceed 11.
C-OP MISS.	The operand C of the PERCENT/RATIO keyword is missing.
C-PICT.END	This message may occur for any of the following conditions: ■ The operand C picture of the PERCENT/ RATIO keyword is not terminated by a right parenthesis. ■ The last character of the picture is the decimal point. ■ The last character of the operand C picture of the PERCENT keyword is not a percent sign (%). ■ Separating characters were found in the operand C picture of the PERCENT/RATIO keyword.
CADR>PGW	The positional indicator of the operand C of the PERCENT/RATIO keyword is outside the print line width.
CADR WRONG	The positional indicator of the operand C of the PERCENT/RATIO keyword is not a 3-digit number. Leading zeroes must be supplied.
CDEC UNNUM	Nonnumeric characters were found in the operand C picture of the PERCENT keyword.
MORE THAN 3	Only three keywords of this type are allowed per report.
NO.NOT NUM	The value in the operand of the TITLE LINENO keyword is not one or two digits.

Message	Explanation
NOMORE CORE	Too many PERCENT, RATIO, or ROUND calculations have been specified. Core is exceeded.
OP.2 MISS.	The second positional indicator is missing for the keyword used.
OP.3 WRONG	SIZE in the LINE SIZE keyword is not followed by a delimiter or MOVEPAGENO.
PARAMREPEAT	Only one parameter of this type is allowed per report.
SHORT LINE	The reduced line size specified in the LINE SIZE keyword is not long enough to contain the page number.
SIZE L1<L2	The modified page width must be less than the input page width.
SZ.1>PGW	The input page width is greater than the print line width.
SZ.1 WRONG	The input page width operand is not a 3-digit number in the LINE SIZE keyword. Leading zeroes must be supplied.
SZ.2>PGW	The modified page width is greater than the print line width.
SZ.2 WRONG	The modified page width operand is not a 3-digit number in the LINE SIZE keyword. Leading zeroes must be supplied.

PR Statement - Processing and Record Selection

The PR statement describes the selection, calculation, and processing of a request. The PR statement allows you to extract and process the information for a report or a subfile or for further processing in subsequent requests.

Position	Entry
01-08	Request name
09-10	Statement type
11-13	Statement sequence
14	Logic level
15	Connector
16	Qualifier A
17-24	Field name A
25-26	Operation
27	Qualifier B
28-58	Field name B, constant, or branch field
59	Result/operand C qualifier
60-67	Field name C/result
68-72	Partial field
68-69	Partial field starting character
70-71	Partial field number of characters
72	Operand A, B, or C

Request name (01-08)

Required entry

Identifies this statement and others accompanying it, if any, as one request. Enter the same name for all statements in the request.

Statement type (09-10)

Required entry

Identifies this as the body of a processing and record selection statement. Enter PR.

Statement sequence (11-13)

Indicates the processing order for statements. See general Rule I.

Entry	Result
Blank	Order of statement input is assumed.
Up to 3 alpha, numeric, or alphanumeric characters	Order in which tasks are performed. Sequencing is in ascending order.

Logic level (14)

Number specifies the sequence of processing multiple AND/OR conditions in a logical expression. The number is similar to using parentheses in an algebraic expression. When logic levels are the same or not specified, AND takes precedence so that A and B or C is always interpreted as (A and B) or C. This entry must be blank for all segment operators.

Entry	Result
Blank	Level of 0 is assumed.
0–9	Number of pairs of parentheses surrounding the condition if expressed algebraically.
Delimiter	GSI CALL statement.

The format of the GSI CALL statement is different from other VISION:Builder formats. Positions [15-71] are reserved for subroutine calls. Enter the keyword CALL, followed by the reusable module name, followed optionally by one or more parameters. Continuations are specified by any nonblank character in [72]. Continuation lines contain a delimiter in [14] and continued parameter list in [15-71]. Refer to the *VISION:Builder Reference Guide* for more information.

Connector (15)

Indicates the relationship between two or more statements within a logical expression. The first line connector must specify a blank connector. This entry must be blank for all segment operators. For processing, enter one of the following codes:

Entry	Result
Blank, 0, or O	OR relationship between statements, segment operators.
A	AND relationship between statements.

Operand A (Qualifier, Field name A)

Identifies the field compared or processed. When using a table lookup, this entry identifies the input argument. When using an RE or RG operator, specify the name of the field whose value is to be used to find a record. For segment operators, this entry is required and identifies the file and segment name for which the segment operation is to be performed.

- Must be blank if the RD operator is specified.
- With scan operators (SL, SR, SN, SS), this entry identifies the field scanned.
- With the combine operator (Cn), it identifies the field to which another field is appended.
- When segment operators are used, this entry specifies the file and segment name.
- In array processing, operand A field name must be blank when using array operators in the statement.

Qualifier A (16)

Identifies the type of field and file where the field exists. The qualifier must be blank if an RD operator is used. The qualifier must be blank, N, O, 0, or 1–9 if an FF or FL operator is used. If an RE/RG operator is used, qualifier A must not be the same as qualifier B.

Code	Location of Field
Blank or N	New master file.
O or 0	Old master file.
1–9	Coordinated files 1–9.
T	Temporary field.
F	Flag.
X	Transaction file.❹
W	Working storage.
V	Linkage section.
A, B, E, G, H, J, K, M, Q, 1–9	Array (must match the qualifier listed on the RF statement).

Field name A (17-24)

This entry specifies one of the following:

- The name of the field being compared or processed.
- The name of the segment as it appears on the LS statement within the file definition for which the segment operation is to be performed.

Enter the field name from the glossary, temporary field, flag, or segment name.

Operand A field cannot be a variable length field (type V), floating point field (type E), or an implicitly defined temporary field for ED operators.

Operand A field must be either a character string (type C) or variable length (type V) field for scan or combine operators.

For generic and duplicate key processing, this is the field whose value is to be used as a search argument for the RE or RG operator.

Operation (25-26)

Required entry

Specifies the operation performed by the data in operand A. Single operators can be in either column. Enter a code from one of the following lists (Logical and relational, Arithmetic, Rounding, Replacement, Branching, and Data access control operators).

Logical and relational operators

Logical and relational operators result in select (true) or nonselect (false) conditions. If the condition is met (true), the data in this field is selected and processing continues with the next PR statement or at the location specified by the GS operator immediately following the logical or relational operation. If the condition is not met (false), no data is selected and the processing either terminates for the request or branches to the location specified by the NS operator immediately following the logical or relational operator.

Code	Condition to be Met
EQ	Operand A equal to operand B.
NE	Operand A not equal to operand B.
GT	Operand A greater than operand B.
GE	Operand A greater than or equal to B.
LT	Operand A less than operand B.
LE	Operand A less than or equal to B.
CV❹	Character validation is to be done against the field in operand A using the pattern specified in operand B.

Code	Condition to be Met
DV❹	Date validation is to be done against either operand A or operand B, using criteria specified in M4SFPARM.
SR	Scan the character field specified in operand A, beginning at the rightmost position, to determine whether it contains the scan argument (character only) specified in operand B.
SL	Scan the character field specified in operand A, beginning at the leftmost position, to determine whether it contains the scan argument (character only) specified in operand B.
SN	Scan the character field specified in operand A to determine if it does not contain the character data specified in operand B.

Arithmetic operators

Arithmetic operators require the specification of all three operands.

Code	Operation
+	Add the contents of the field in operand A to those of operand B and put the results in operand C.
-	Subtract the contents of the field in operand B from those of operand A and put the results in operand C.
*	Multiply the contents of the field in operand A with those of operand B and put the results in operand C.
/	Divide the contents of the field in operand A by those of operand B and put the results in operand C.

Rounding operators

Rounding operators require the specification of all three operands, except RR, which requires the specification of only two operands.

Code	Operation
+R	Add the contents of the field in operand A to those of operand B and put the rounded results in operand C.
-R	Subtract the contents of the field in operand B from those of operand A and put the rounded results in operand C.
*R	Multiply the contents of the field in operand A with those of operand B and put the rounded results in operand C.
/R	Divide the contents of the field in operand A by those of operand B and put the rounded results in operand C.
RR	Replace the value from either operand A or B (only one can be specified) and put the rounded results in operand C.

- Variable length fields cannot be used with any of the rounding operators.
- Operand C cannot be a character or floating point field type when using any of the rounding operators.
- When using replace with rounding from a character field to a zoned field, the sign will always be C for a positive value and D for a negative value. Note that replace without rounding from a character field to a zoned field produces an F sign for a positive value and D for a negative value.
- Rounding only occurs if the number of decimal places in the result field is less than the number in the computed replacement value.

Replacement operators

Replacement operators move the contents of one operand into another operand.

Code	Operation
R	Replace the value from either operand A or B (only one can be specified) into operand C.
Cn	Link operands A and B (inserting the specified number (n) of blanks (0–9) between them) and place the new character string into operand C.
ED	Edit the contents of operand A according to the edit pattern provided in operand B and place the edited result into operand C.
SS	Scan operand A using the argument in operand B and replace it with the value specified in operand C. B can be a character string, character field, or a pattern; C must be a character string field.

Code	Operation
TL	Search the table specified in operand B for an argument equal to that specified in operand A and place the result in operand C.
TN	Search the table specified in operand B for an argument equal to or nearest to that specified in operand A and place the result in operand C.
TB	Search the binary search table specified in operand B for an argument equal to or next larger than that specified in operand A and place the result into operand C.
TS	Search the binary search table specified in operand B for an argument equal to or next smaller than that specified in operand A and place the result into operand C.
TI	Search the binary search table specified in operand B for a linear interpolated value and place that value into operand C.
JL❹	Left-align the data in the field specified in operand A or B (only one can be specified), eliminate leading blanks, and place the result into operand C.
JR❹	Right-align the data in the field specified in operand A or B (only one can be specified), eliminate trailing blanks, and place the result into operand C.

- For ED operations, operand A cannot be a type V, E, or implicit temporary field.
- For ED operations, static partial field specifications can be used for operand A if type C. If type C, the edit pattern must use the system delimiter character in the picture string; all other characters are edit insertion characters. If a numeric field, the edit pattern can use the system delimiter picture string or the COBOL-like edit symbols as defined in M4PARAMS in the picture string. See the output specification Rn statement output edit entry [53-67] for details on edit patterns. Time processing fields (type L and S) are edited as packed decimal fields. If a Date field (type D), the edit pattern is a date format picture, for example MM/DD/YY. Refer to the *IBM Language Environment Reference Manual* for a list of valid picture characters. The picture is used as a parameter in a call to CEEDATE.
- For ED operations, operand B field must contain a picture string as currently allowed in the output edit field of the Rn statement. The maximum number of digits in the edit pattern for a numeric field is 30.
- For ED operations, operand C must be a type C field. It can be an implicit temporary field.
- Dynamic partial fielding is not allowed on any field for ED operations.

Branching operators

Branching operators deviate from the normal order of statement processing (ascending order by sequence number). There are two types of branching:

- *Forward branching* - allows processing to bypass one or more PR statements and process statements elsewhere in either the same request or the beginning of a later one.
- *Backward branching* - allows processing to return to the beginning of the current or previously processed request for another pass.

Branching	Operator Conditions
NS	Conditional branch to another statement, REQUEST request name, OUTPUT, SUB request name, RETURN, TYPE n, END, NEXT MASTER, or NEXT REQUEST. Conditional: does not meet the conditions set forth in the previous processing statement, such as EQUAL or NOT EQUAL (false situation).
GS	Conditional branch to another statement, REQUEST request name, OUTPUT, SUB request name, RETURN, TYPE n, END, NEXT MASTER, or NEXT REQUEST. Conditional: meets the conditions set forth in the previous processing statement (true situation).
GO	Unconditional branch to a specific place, such as a statement, REQUEST request name, RETURN, END, OUTPUT, NEXT MASTER, NEXT REQUEST, TYPE n, or SUB request name. Unconditional: takes the branch regardless of the results of the previous processing statement.

- The GS and NS operators are mutually exclusive.
- An implied NS END is assumed if a GS or NS operation is not coded following logical or relational operators.
- An NS or GS operator can be coded following array, segment, and some read operators. See Chapter 3, *Flags* for a description of the flags ASTATUS, LSTATUS, and RSTATUS, respectively.

Data access control operators

Data access control operators control access to records at certain times when processing requests (see Request type).

With the data access array, segment, and read operators, status codes indicating the conditional results of the operations are set in the ASTATUS, LSTATUS, and RSTATUS flags, respectively. Additionally, an NS or GS operator can be coded following the data access operators (except RD). See Chapter 3, *Flags* for descriptions of the flags.

Code	Operation
RD	*Read operator* - sequentially reads the coordinated file specified in qualifier B (operands A and C are not allowed). R must be specified on the RF statement [53].
RE	*Read key equal to* - directly reads the coordinated file specified in qualifier B. This operation reads the first record whose full or high order key field value is equal to the value specified in operand A. G or R must be specified on the RF statement [53].
RG	*Read key greater than or equal* - directly reads the coordinated file record specified in qualifier B whose full or high order key value is greater than or equal to the value specified in operand A. G or R is specified on the RF statement [53].
RS	*Release segment operator* - releases a segment type (specified in operand A) previously obtained with a segment operator (specified in operand A).
FS	*Find segment operator* - selects a given segment by its key if operand B is specified. If operand B is not specified, the segment following the one most recently retrieved within the current parent is retrieved. Segment looping is suppressed on that segment type although looping of parent segments occurs normally.
FF	*Find first segment operator* - obtains the first occurrence of the segment named in operand A for processing. (This operator cannot be used with memory optimized (MOSAIC) processing for SQL or GDBI files.)
FL	*Find last segment operator* - obtains the last occurrence of the segment named in operand A for processing. (This operator cannot be used with memory optimized (MOSAIC) processing for SQL or GDBI files.)
LA	*Locate array operator* - releases control of a specific data cell after it is located using the LD operator. This makes the entire array available for automatic looping. (Reference to a field in the array starts the loop.)

Code	Operation
LC	*Locate column operator* - locates the column number specified in operand C. Subsequent reference to a field in this array will start the loop through the rows in the column, extracting information for processing requests.
LD	*Locate data cell operator* - locates the specific data cell at the intersection of row number in operand B and column number specified in operand C.
LR	*Locate row operator* - locates the row specified in operand B. Subsequent reference to a field in this array will start a loop through the columns in the row, extracting information for request processing.

- The RE or RG operator used in conjunction with the RD operator can be used to retrieve duplicate-keyed root segments or to perform a generic search.
- When using RE or RG operators in conjunction with the RD operator to retrieve duplicate-keyed root segments, only the operand B qualifier [27] is used. It specifies the coordinated file being read (1-9). R is specified on the RF statement [53].
- An RD operation is bypassed if it occurs after a reference to a lower level segment for that file within the same request.
- When using an RE or RG operator and operand B is blank, the generic key search field used defaults to the field specified on the RF statement. G is specified on the RF statement [53].
- The RE or RG operators are only allowed on VSAM, DL/I, or relational user controlled coordinated files.
- Operand B also specifies a value or field in conjunction with a find segment (FS) operation.

Qualifier B (27)

Identifies the type of field and location of the file where the field exists. This qualifier must be P if specified as an ED operator. This qualifier must be blank if specified as FF and FL segment operators.

Code	Location of File
Blank or N	New master file field.
O or 0	Old master file field.
1–9	Coordinated files 1–9 field.
T	Temporary field.
C	Character string constant field.

Code	Location of File
D	Decimal constant field.
F	Flag field.
$	Time constant field.
X	Transaction file field.❹
L	Table definition (table name) field.
W	Working storage.
V	Linkage section.
A, B, E, G, H, J, K, M, Q, 1–9	Array field (must match qualifier listed on RF statement).
P	Scan/Validation pattern or edit pattern.

Field name B, constant, or branch field (28-58)

Operand B can be a field, constant, edit pattern, validation pattern, scan pattern, table name, generic field name, or branch location.

Qualifier P defines an edit pattern used with the ED operator. Partial field specification is not allowed. See the output specification Rn statement output edit entry [53-67] for details on edit patterns.

With a Generalized System Interface (GSI), constants (name of a temporary field, working storage field, or character decimal constant) are used as the comparative value in the constructed segment search argument and are converted to the type, length, and scale of the operand A search or key field.

With the combine operator, field name B identifies the field combined with field name A.

Qualifier P defines a validation pattern (CV operator), scan pattern (scan operators SL, SR, Sn, and SS) or edit pattern for numeric fields (ED operator).

Symbol	Character Set Represented
9	Numeric (0–9).
a	Alpha (A–Z, a–z) or blank.
A	Alpha (A–Z) or blank.
B	Blank.
C	No validation.
D	System Delimiter.
I	Numeric (0–9) or blank.
x	Alpha (A–Z, a–z), numeric (0–9), or blank.
X	Alpha (A–Z), numeric (0–9), or blank.
y	Alpha (A–Z, a–z) or numeric (0–9).
Y	Alpha (A–Z) or numeric (0–9).
z	Alpha (A–Z, a–z).
Z	Alpha (A–Z).
Literals	Any character(s).
User defined	Provided by you at system installation time.

- A minus sign (-) before any of these characters means scanning for other than the specified validation symbol. (One minus sign operates on only one validation symbol or literal. For example, -99 represents a nonnumeric character followed by a numeric character, not two nonnumeric characters.)
- System delimiters must be used to surround literals in a scan pattern.
- A packed field in operand B cannot exceed a packed length of 8 (15 digits) for multiply or divide. If this length is exceeded, a type 5 message is issued after final decode and the run terminates.

Rules for constants

When using constants, the following rules apply.

- Constants can appear only in operand B. Enter the constant value left-aligned. Any characters recognized by the hardware can be used (except the system delimiter). A character string, identified by the C qualifier, can be up to 31 characters long including embedded blanks. Leading blanks are significant in character string constants. If this field is numeric, it cannot exceed 15 digits. Scientific notation can also be used.
- The D qualifier restricts the field to numeric data. A decimal constant can be an integer or a number containing an explicit decimal point. Its size is restricted by hardware single precision operations. Any decimal constant is converted to a field with attributes compatible for the operation. These attributes are usually those of the leftmost defined field except that it cannot exceed a packed length of 8. Scientific notation cannot be used.
- The $ qualifier identifies the constant as a time value in hours-minutes-seconds format (sign hh...h:mm:ss.nnn). A sign can be specified. Decimals of seconds can also be specified. The colon (:) separating the hours, minutes, and seconds is an installation parameter.
- The P qualifier identifies field name B as a scan/validation pattern for the CV or scan operators.
- The P qualifier identifies field B as an edit pattern for the ED operator. If operand A is a character field (type C), the edit pattern must use the system delimiter character in the picture string. All other characters are edit insertion characters.

  If operand A is a numeric field (type Z, P, F, L, or S), the edit pattern can use the system delimiter picture string or the COBOL-like edit symbols as defined in M4PARAMS in the picture string. The M4PARAMS edit symbols are the digit select, zero suppress, check protection, decimal point, grouping, currency, plus, and minus characters. The maximum number of digits in the edit pattern for a numeric field is 30.
- If a constant is specified in a find segment operation, the value entered causes VISION:Builder to find the specified segment whose key matches it. If operand B is not specified, the segment following the one most recently retrieved within the current parent is retrieved.

Type of Processing/ Operation	Entry
Normal	Field name from glossary, temporary field, flag name, or constant.
Table	Table name from glossary.
LC, LA	No operand B entry.

Type of Processing/ Operation	Entry
LR, LD	Numeric value of row to be located.
Scan (SL, SR, SN, SS)	Scan argument or validation pattern for scan.
FS	Field name being used to locate specific segment whose key matches this value (can be left blank).
FF, FL	Blank.
RD	Blank.
RE, RG	Blank - defaults to generic key specified in the RF statement. This entry must be blank if R is specified in position 53 of the RF statement. Generic key field name used to override the generic key specified in the RF statement.
ED	Edit pattern.

Type of Processing/ Operation	Entry
Branching (GO, NS, GS)	*TYPE n* - branches to the beginning of type n request processing, where n is 1, 2, or 4. Sequence number - branches to a statement having a higher sequence number of the same request. b/20 and 020 are not the same. *OUTPUT* - branches to the reporting sections of the same request. *END* - branches around all subsequent PR statements and all reporting sections in the same request but stays within all indicated loops. If there are no initiated loops, it branches to NEXT REQUEST. *NEXT REQUEST* - branches to the next request. *NEXT MASTER* - bypasses all following requests and reads the next master file record, operating as if a GO RETURN were issued from a subroutine. *REQUEST* request name - branches to any specified request of the same type in the run. In place of request name, enter the name of the request you want to branch to. *SUB* request name - invokes the subroutine named in request name. *SUB* set name - invokes the subroutine request set named in set name. *RETURN* - terminates this subroutine request or set. It returns to the statement following the one that called the subroutine request or request set. If issued in any other type of VISION:Builder request, the RETURN branch returns control to VISION:Builder internal flow, effectively terminating processing of the current request type for this pass through the request chains.

Result/operand C qualifier (59)

Location of field where result/operand C resides. This qualifier must be blank for all segment operators.

Code	Location of Field
Blank or N	New master file field.
O or 0	Old master file field.
1–9	Coordinated files 1–9 field.
T	Temporary field.
F	Flag field.
X	Transaction file field.❹
W	Working storage.
V	Linkage section.
A, B, E, G, H, J, K, M, Q, 1–9	Array field (must match qualifier on RF statement).

Field name C/result (60-67)

Names the field where the result of an operation is stored for further computation or output. See general Rule A. Field name C/result contains the field to be substituted (type V or C only) in scan operations.

When using the ED operator, only static partial fielding is allowed (no dynamic).

Operand C can be an implicitly defined temporary field. The attributes assigned to the implicit temporary field are type C field with a length equal to the number of characters in the edit pattern.

- Entering a field name from the new master file work area causes a permanent change in the new master file field value.
- Result/operand C cannot be a count field on the file definition.

Partial field (68-72)

Partial field entries (starting character, number of characters, operand A, B, or C) allow processing of any part of a character string or type V field in the operand specified in [72]. Partial field entries are not permitted for the FF and FL operators.

Both starting characters [68-69] and number of characters [70-71] must be specified for character string fields if using partial fielding.

For type V fields, the number of characters can be left blank; the remaining length of the field will be used. Partial fielding is not allowed with RE and RG operators. Both starting and number of characters must fall within the range of the operand specified.

Partial field starting character (68-69)

Location of first character used in the partial field.

Entry	Result
Blank	No partial fielding.
1–99	Position of character beginning the partial field.
LS	Current value of left start flag.
LN	Current value of left number flag.
MS	Current value of middle start flag.
MN	Current value of middle number flag.
RS	Current value of right start flag.
RN	Current value of right number flag.

Partial field number of characters (70-71)

Number of characters forming the partial field.

Entry	Result
Blank	No partial fielding.
1–99	Number of characters that form the partial field (up to the maximum length of the field).
LS	Current value of left start flag.
LN	Current value of left number flag.
MS	Current value of middle start flag.
MN	Current value of middle number flag.
RS	Current value of right start flag.
RN	Current value of right number flag.

Operand A, B, or C (72)

Identifies the operand to which the partial field specifications apply.

Entry	Result
Blank	Leftmost operand with an entry in it.
A	Operand A.
B	Operand B.
C	Operand C.
Any nonblank character	Continuation character for GSI CALL statements.

Qn Statement - Extended Output Edit

The Qn statement contains an output edit pattern up to 30 positions long.

Position	Entry
01-08	Request name
09-10	Statement type
12-14	Extended Output Edit Identifier
15-44	Extended Output Edit Pattern

Request name (01-08)

Required entry

Identifies this statement and others accompanying it (if any) as one request. See general Rule A. Enter the same request name for all statements in the request.

Statement type (09-10)

Required entry

Identifies a Qn statement. Enter Qn.

Extended Output Edit Identifier (12-14)

The 3-position character value is used to link the Qn statement with an Rn Output Specification statement.

Extended Output Edit Pattern (15-44)

Enter an output edit pattern, up to 30 characters long, to be used to edit a field for output on a report. This edit pattern can consist of either of two styles: the delimiter style or the COBOL style. The delimiter style pattern can be used when editing character data or numeric data. The COBOL style pattern can only be used with numeric data. Do not use this entry with numeric field type E. For information about requirements for output editing of character string data, numeric data, and variable length fields, see Character string data on page 2-194, Numeric data (P, Z, and F only) on page 2-195, and Variable length fields (type V fields) on page 2-195.

RA Statement - Run Control Group

The RA statement gives the ability to overdefine an array with a secondary definition, allowing the same array to be referenced in multiple ways.

Position	Entry
01-08	Run name
09-10	Statement type
11	Primary qualifier
12	Secondary qualifier

Run name (01-08)

Required entry

Identifies this statement as belonging to a particular run group. Enter the same name as entered on the RC statement.

Statement type (09-10)

Required entry

Identifies a run parameter statement. Enter RA.

Primary qualifier (11)

Required entry

Entry	Result
A, B, E, G, H, J, K, M, Q, and 1–9	Enter a valid array qualifier to be the primary array definition.

Note: The same qualifier can appear on more than one RA statement.

Secondary qualifier (12)

Required entry.

Entry	Result
A, B, E, G, H, J, K, M, Q, and 1–9	Enter a valid array qualifier to be the secondary array definition.

- Secondary qualifiers cannot be reused as primary or secondary qualifiers on any other RA statements.
- The record length of each secondary array must be less than or equal to the record length of the primary array.
- An RF statement must exist for any array referenced on an RA statement and it must precede the RA statements in the M4INPUT stream.
- For secondary arrays specified on an RA statement, there must be a valid RF statement, where the label and DTF/ddname are blank.
- Primary array definitions can be either external or internal arrays. Secondary array definitions must be internal arrays.
- Cell size times the number of columns for a secondary array must exactly equal the cell size times the number of columns for its primary array.

RC Statement - Run Control Group

The run control statement (RC statement) specifies the master file, report files, and reject and audit transaction files[4] for a run. It also allows you to control run dependent parameters. The RC statement must be the first input statement in a run.

Position	Entry
01-08	Run name
09-10	Statement type
11-18	Master file glossary name
19	Old in
20	New out[4]
21	Sequence
22	Direct read?[4]
23	Update-in-place?[4]
24	Transaction file in[4]
25	Report file in/out
26	Audit file out[4]
27	Single buffer all files
28	Sort control

Position	Entry
29	Override delimiter
30	Scan/terminate control
31	Source statement listing
32-33	Spool blocking factor
34	Rejected transactions[4]
35-37	Resource optimization
35	Storage?
36	Message processing?
37	Report file
38	Source statement out
41-56	Start search
57-72	End search

Run name (01-08)

Required entry

Identifies a particular run. See general Rule A. This name must be the same on all run control group statements in the run. The system delimiter or the character specified in position [29] of this statement cannot be used.

Statement type (09-10)

Required entry

Identifies a run control statement. Enter RC.

Master file glossary name (11-18)

Identifies the master file for this run.

Enter the name as it appears on the file definition glossary from the FD statement [01-08] or the run data group name as it appears on the DB statement.

Old in (19)

Indicates that an old master file is to be read. Enter one of the following:

Entry	Result
Blank	No old master file in.
S	Standard labels.
U	Unlabeled.
X	Unlabeled, but no rewind on OPEN or CLOSE.
N	Nonstandard labels. If this parameter is used in the OS/390 environment, you must supply own-code or system generation functions for VISION:Builder to process; otherwise, VISION:Builder assumes standard labels.
V	Unlabeled, multi-volume tape input (VSE only). If specified, operator intervention is required at end of volume to determine EOV or EOF. If specified for any other system, VISION:Builder treats it as unlabeled (U).

OS/390 users must use one of these codes, but the label parameter must be used in the JCL.

Enter S for a relational table that is to be read as an old master file.

New out (20)

Indicates that a new master file is output. This entry must be blank for IMS files for retrieval only. Enter one of the following:

Entry	Result
Blank	No new master file out.
S	Standard labels.
U	Unlabeled.
X	Unlabeled, but no rewind on OPEN or CLOSE.
N	Nonstandard labels. If this parameter is used in the OS/390 environment, you must supply own-code or system generation functions for VISION:Builder to process; otherwise, VISION:Builder assumes standard labels.

Note: OS/390 users must use one of these codes, but the function can be achieved only by JCL.

Sequence (21)

Specifies the sequencing of the input master file.

Entry	Result
Blank	The master file is checked for ascending sequence errors.
U	The master file is not checked for sequence errors. No transaction file is permitted. For an IMS HDAM file, if direct-read is not specified, VISION:Builder assumes a U.
E	The master file is checked for sequence errors, but equal keys are not regarded as errors. Transaction records[4] and coordinated files are applied to the first record of those with equal keys.
P	No sequence checking is performed and records are retrieved in physical sequential order. No transaction or coordinated files are permitted. This specification applies to key sequenced VSAM files only (type K).

Direct read?[4] (22)

Applies only to master files that are read randomly, based on the transaction records, and have been created using an access method that allows direct-read (for example, IMS, relational, VSAM, or ISAM).

Entry	Result
Blank or N	The master file is processed sequentially and sequence checking is performed.
Y	Sequence checking is not performed. During transaction processing, an IMS master file, relational file, key sequenced VSAM file, or ISAM master file is accessed directly by key value. In this mode of operation, a transaction file must exist and only those records having transactions are accessed for processing.

Note: For IMS files, if the run control statement specifies an HDAM master file and the sequence entry does not specify a U, a type 1 warning message will be issued stating the file will be processed as unsequenced.

Update-in-place?❹ (23)

Enter one of the following:

Entry	Result
Blank or N	The master file is processed as read only.
Y	An updated record occupies the same physical position on a direct access device as the original old master record. IMS files are valid for update.

- Do not specify a new master file when Y is entered. However, the data stored on the physical device comes from the new master file area in storage.
- A master file can be updated-in-place only if the storage medium is a direct access storage device. Record length can be changed only for variable length key sequenced VSAM and ISAM files.
- An entry of Y for a relational file indicates that the master file is to be updated. It must also be a memory optimized (MOSAIC) file for updating-in-place to be allowed, when the file contains a hierarchy (a root segment plus one or more dependent segments). If the file is defined as a single table view (root segment only), MOSAIC is not required.

The following table shows possible combinations of the direct-read, update-in-place, and new master output entries.

ENTRY	Direct-read?	Y	N	Y	Y	N	N
	Update-in-place?	Y	Y	N	N	N	N
	New master output?	N/A	N/A	N	Y	N	Y
CAPABILITY	Update master records	A	A	N/A	A	N/A	A
	Create master records and/or insert segment occurrences in existing master records	A	A	N/A	A	N/A	A
	Delete master records	A	A	N/A	A	N/A	A
	Transaction file	R	A	R	R	A	A
	Reports	A	A	A	A	A	A
	Create or update new master only in core	N/A	N/A	A	N/A	A	A
KEY	N/A = not allowed R = required N = not specified		Y = specified A = allowed				

Note: When the Direct-read is set to N and the Update-in-place is set to Y, then the update-in-place for sequential master files can only be used to modify existing master file records.

Transaction file in❹ (24)

Indicates the existence of a transaction file.

Entry	Result
Blank	No transaction file.
S	Standard labels.
U	Unlabeled.

Entry	Result
X	Unlabeled, but no rewind on OPEN or CLOSE.
N	Nonstandard labels. If this parameter is used in the OS/390 environment, the reading of nonstandard labels must be done by own-code or system generation functions. If these are not used, VISION:Builder assumes standard labels.
V	Unlabeled, multi-volume tape input (VSE only). If specified, operator intervention is required at end of volume to determine EOV or EOF. If specified for any other system, VISION:Builder will treat it as unlabeled (U).

Note: OS/390 users must use one of these codes. If U, X, N, or V is specified and this file is output to tape, the label parameter must be used in the JCL.

Report file in/out (25)

Indicates the existence of a report file. Enter one of the following:

Entry	Result
Blank	No report file.
S	Standard labels.
U	Unlabeled.
V	Unlabeled, multi-volume tape input (VSE only). If specified, operator intervention is required at end of volume to determine EOV or EOF. If specified for any other system, VISION:Builder will treat it as unlabeled (U).

- OS/390 users must use one of these codes, but the function can only be achieved by JCL. Report phase input is assumed if no other files are specified on the RC statement.

Audit file out (26)

Specifies the writing out of a sequential file for deleted master file records. Enter one of the following:

Entry	Audit File
Blank	No audit file.
S	Standard labels.
U	Unlabeled.
X	Unlabeled, but no rewind on OPEN or CLOSE.
N	Nonstandard labels. If this parameter is used in the OS/390 environment, the writing of nonstandard labels must be done by own-code or system generation functions; otherwise, VISION:Builder assumes standard labels.

- OS/390 users must use one of these codes. If U, X, or N is specified and the file is output to tape, the label parameter must be used in the JCL.
- This entry must be blank for relational master files.

Single buffer all files (27)

Limits buffering of files. Enter one of the following:

Entry	Result
Blank or N	Input and output files use installation standard buffering (see M4PARAMS in the *VISION:Builder for OS/390 Installation Guide* or the *VISION:Builder for VSE Installation Guide*).
Y	Input and output files are single-buffered.
D	Input and output files are double-buffered.

- Single-buffering does not apply to VSE unit record devices.
- OS/390 users can achieve the single-buffering capability with JCL.

Sort control (28)

Specifies sort control statement generation. This entry must not be used when producing a graph. Enter one of the following:

Entry	Result
Blank or N	The report file must be sorted before the report is printed. VISION:Builder generates sort control statements. Depending on your system, you must provide the appropriate JCL, system commands, RTPs, or CMS commands for the sort program used.
X	A report file is generated but sort control statements are not produced and the report step is not entered.
Y	No sort is required. A Y entry can be used only when there is a single report file not requiring sorting.
1–9	Specifies the number of sort work files. These entries apply only to VSE. The number can be omitted, in which case SORT WORK=DA is generated.

- OS/390 - the report step is entered without returning to job control. JCL is needed only for the output report file. If M4LIST is not a SYSOUT data set or a printer, DISP=MOD must be specified; otherwise, the report step output will be written over the decode processing step output.
- CMS - the report step is entered without returning to M4EXEC. If M4LIST output is being written to disk, DISP=MOD must be specified for M4LIST; otherwise, the report step output will be written over the decode/processing step listing.
- If single-step processing (R in position [37]), all I/O to M4REPO/M4REPI is eliminated, along with the reactivation of VISION:Builder. This eliminates the report file EXCPs and saves processing and I/O time.
- VSE - JCL is needed for both M4REPO and M4REPI (for three-step processing only) and should define the same data area. This entry can be used only when there is a single report that does not require sorting.

Override delimiter (29)

Identifies the delimiter used in the run.

If blank, VISION:Builder uses the installation standard delimiter (see M4PARAMS in the *VISION:Builder for OS/390 Installation Guide* or the *VISION:Builder for VSE Installation Guide*). Enter any character except an underscore(_) or tilde(~).

Scan/terminate control (30)

Sets the conditions under which a run can be terminated. Enter one of the following:

Entry	Result
Blank	Master file is passed and all valid requests are processed.
S	Scan only run. The requests are decoded. Cataloged request maintenance is performed and diagnostics issued, but no processing is done.
T	Run is terminated if decoding errors are found in any request in the run.
R	A scan only run is performed and a sample report is generated for each valid request in the run. Sample reports allow you to check the appearance of your data without actually preparing test data.
M	Scan only run for requests that are to be decoded according to mapping decode rules. Cataloged request maintenance can be performed and diagnostics issued, but no processing is done. G
N	Scan only run for requests that are to be decoded according to mapping decode rules. A sample report is produced for each valid request in the run. G
4	Normally, a type 4 message terminates a run. Specifying a 4 in scan terminate permits the run to continue; however, errors can be bypassed and the results of subsequent processing can be invalid.

Note: The scan run allows you to check your requests, as well as run control group statements.

Source statement listing (31)

Determines the type of source statement listing.

Entry	Result
Blank or N	Format and print source statements.
Y	Suppress source statement listings and PAL reports.
A	Suppress all output to M4LIST except for reports, type 3 (or higher) messages and instream definitions. Suppress VSE sort SYSLST. Suppress PAL reports.
B	Suppress source statement listings, PAL reports, and signon page.
X	Suppress source statement listings and PAL reports. Produce an 80-80 listing showing output to M4SSOUT. (Valid only in source statement retrieval runs.)
C	Provide a complete listing of both application and mapping request statements. G
D	Output source statement listings, diagnostics, statistics, and signon page to M4LIST. Output report(s) and report START, END, NODATA, NUMBER OF LINES OF PRINT, NUMBER OF INVALID/MISSING/OVERFLOW FIELDS, and INCOMPLETE SUMMARY to M4LIST1. Ignore alternate M4LIST specifications on En statements.
E	Perform both C and D. G
P	Suppress printing of PAL reports and all cataloged requests, including cataloged mapping requests.
Q	Suppress printing of all nonmapped cataloged requests and PAL reports.
R	Suppress printing all cataloged requests and direct output to alternate M4LIST1. Suppress PAL reports.
S	Suppress printing of all nonmapped cataloged requests and direct output to alternate M4LIST1. Suppress PAL reports.

- You must provide JCL for M4LIST1.
- If nonmapped requests are suppressed, the associated CR statement will still print. If mapped requests are suppressed, both the CR statements and the requests will be suppressed.

Spool blocking factor (32-33)

Valid for VSE only

Permits blocking of print image lines.

Entry	Result
1–99	Equal to the blocking factor of the print image output when assigned to a device other than a unit record device. Must be right-aligned.

- OS/390 users achieve this capability with JCL.
- Using this capability presupposes adequate storage for buffers. The output is blocked and the leading character is assumed to be an ASA control character for VSE systems. The following ASA characters are used in VISION:Builder.

Blank	Single-space
0	Double-space
-	Triple-space
1	Skip to channel 1

Rejected transactions[4] (34)

Specifies output to a file of the transactions rejected during an update run.

If direct-read [22] is specified and all transactions for a record are rejected, the results are as follows:

- Normal requests are not executed for the record.
- If update-in-place is also specified, the record is not rewritten.
- If a new master file is created, the record is not written to it.

Entry	Result
Blank	No rejected transaction file.
S	Standard labels.
U	Unlabeled.
X	Unlabeled, but no rewind on OPEN or CLOSE.
N	Nonstandard labels. If this parameter is used in the OS/390 environment, the writing of nonstandard labels must be done by own-code or system generation function. If these are not used, VISION:Builder assumes standard labels.

- When S, U, X, or N is entered, the rejected transaction records are output with the same format and blocking factor as the transaction input file.
- OS/390 users must use one of these codes. If U, X, or N is specified and this file is output to tape, the label parameter must be used in the JCL.

Resource optimization (35-37)

The resource optimization entries (storage, message processing, report file) give you optional trade-offs (such as storage versus processing time, main storage of messages versus dynamic loading, report file optimization versus producing detail items).

Storage? (35)

Either processing speed or main storage allocation can be selected for optimization. Enter one of the following:

Entry	Result
Blank or N	Optimize processing speed. Extra main storage is utilized to improve processing speed.
Y	Optimize main storage allocation. Maximum conservation of main storage is achieved during processing. An entry of Y is not allowed if the master file is relational and update-in-place.

Message processing? (36)

Indicates whether messages that occur frequently during the processing step of VISION:Builder remain in storage. This eliminates message modules being loaded more than once during processing. Enter one of the following:

Entry	Result
Blank or N	Messages do not reside in storage.
Y	Messages accessed during processing reside in storage for access during that phase. Extra main storage is used to store these messages.

Note: For processing runs with a transaction file specified, a Y should be specified, especially when type 1 transaction rejected messages are output by VISION:Builder.❹

Report file (37)

Specifies a way of reducing I/O processing of report files.

Entry	Results
Blank	Do not optimize the report file. If X is specified in sort control [28], this must be blank.
R	Single-step processing. This is effective for both SORT and NO-SORT types of runs.
S	Report file optimization is wanted. ■ Preliminary summaries for numeric fields are taken during the final output phase of the sort program to reduce the number of records passed to the report step. ■ This is effective for summary only reports that request TOTAL, CUM, PERCENT, or RATIO summaries only on numeric fields (not type C). ■ There can be no more than 40 summary and control fields in the report, and the length of the data record on the report file cannot exceed 400 bytes. ■ This entry is not allowed in a NO-SORT run.

- In OS/390, an M4SRTLIB DD statement that points to the LOADLIB containing the VISION:Builder routine MARKSS must also be included in the sort step JCL.
- In NO-SORT runs (Y in position [28]), you cannot specify S because all I/O to M4REPO/M4REPI is eliminated, along with the re-activation of VISION:Builder. NO-SORT runs do not create report file EXCPs, saving processing and I/O time.

Source statement out (38)

Specifies output of source statements. Required entry for source statement retrieval runs. OS/390 users must use one of the following. If U, X, or N is specified and this file is output to tape, the labels parameter must be used in the JCL.

Entry	Result
Blank	No source retrieval file.
S	Standard labels.
U	Unlabeled.
X	Unlabeled, but no rewind on OPEN or CLOSE.
N	Nonstandard labels.

Note: In the OS/390 environment, the writing of nonstandard labels must be done by own-code or system generation function; otherwise, VISION:Builder assumes standard labels.

Start search (41-56)

Limits processing to only part of the old master file, bypassing records with a record key lower than the one specified if the file is sequential. If the file is not sequential (for example, DL/I, relational, VSAM, ISAM, or non-HDAM), a direct-read is performed. Entry must be blank for HDAM DL/I data sets.

Valid Entries	Description
Blank	Leave blank if the selection criteria of any request in the run applies to the entire file.
Record Key Value	Enter the record key value of the old master file record where the search is to begin. Must be left-aligned.

- Do not use a start or end search if processing an unordered master file.
- Record key must apply only to the first level (major) key field and must be a character string. If the key is longer than 16 bytes, only the first 16 can be entered.
- If the key is numeric, the value will be converted to the corresponding key field type. File processing begins with the key that matches this value or the next higher one if the specified key is nonexistent.
- If the run control statement specifies an HDAM master file for a DL/I data set and the start/end search values are given, the values are ignored and a warning message is issued.
- For relational files, the start/end search is interpreted as part of any LR statements governing the root segment. The LR statement conditions must be true AND the start/end search key value must be acceptable for the record to be processed.

End search (57-72)

Limits processing to part of the old master file. The run is terminated after processing the last record with the key specified or the next higher one if the specified key is nonexistent.

Valid Entries	Description
Blank	Leave blank if processing is to continue on every record until end of file is reached.
Record Key Value	Enter the record key value of old master file record where search is to end. Must be left-aligned.

- Do not use a start or end search if processing an unordered master file.
- Record key must apply only to the first level (major) key field and must be a character string. If the key is longer than 16 bytes, only the first 16 can be entered.
- If the key is numeric, the value will be converted to the corresponding key field type. File processing ends with the record key whose value matches this value or the next higher one if the specified key is nonexistent.
- For relational files, the start/end search is interpreted as part of any LR statements governing the root segment. The LR statement conditions must be true AND the start/end search key value must be acceptable for the record to be processed.

RF Statement - Run Control Group

The additional files statement (RF statement) is used when subfile or alternate report files are generated, coordinated files or array files are used, passwords are required for VSAM files, authorization ID is required for DB2 files, MOSAIC processing is specified for any file, or user controlled I/O is needed for specific files.

Position	Entry
01-08	Run name
09-10	Statement type
11-18	File definition name
19	Label
20-27	Logical file name
28-51	Fields to use as key fields for input files
28-35	Key1
36-43	Key2
44-51	Key3
52	ICF
53	CORD
54	MOSAIC
55-62	User I/O module name
63	Array
65-72	Password authorization ID

Run name (01-08)

Required entry

Enter the same name as entered on the RC statement. See general Rule A.

Statement type (09-10)

Required entry

Identifies a statement that names the additional files being used during a run. Enter RF.

File definition name (11-18)

Required entry

Identifies the file by a name you supply.

Type of File	Entry
Coordinated, Master, Working Storage, and/or Linkage Section	File name as it appears in the file definition.
Transaction	Master file definition name.4
Array	Array name as it appears on the array definition.
Non-IMS Subfiles and/or Alternate Report Files	DTF/ddname or subfile name as entered on En statement for this run.
IMS Subfiles	The DBD name assigned to this file.I
Relational Master File, Coordinated File, or Subfile	File name as it appears in the file definition. In the case of generated tables, enter the same name as entered on the En statement in positions [44-51]D.

Note: When taking IMS checkpoint calls, you cannot use IMS subfiles. I

Label (19)

Specifies the types of labels.

Entry	Result
Blank or S	Standard labels.
U	Unlabeled.
X	Unlabeled, but no rewind on OPEN or CLOSE.
N	Nonstandard labels. If this parameter is used in the OS/390 environment, the reading or writing of nonstandard labels must be done by own-code or system generation functions. If these are not used, VISION:Builder assumes standard labels.
V	Unlabeled, multi-volume tape input (VSE only). If specified, operator intervention is required at end of volume to determine EOV or EOF.

- For the master file (M4OLD or M4NEW), the label entry from the RC statement will be used if different from this entry.
- OS/390 - you must use one of these codes, but the function can only be achieved by JCL.
- ICF - this entry must contain S or blank.
- IMS DL/I VSE - all of the entries shown are valid and provided to allow compatibility with non-IMS VISION:Builder files where they convey labeling conventions.
- DB2 - enter an S or blank.

Logical file name (20-27)

Specifies the VISION:Builder logical name for the file. For nonrelational files, this is also the DTF or ddname for the file. The DTF/ddname for the physical database (as defined in the DBD) can be different from the one supplied here.

Type of File	Entry
Master (old)	M4OLD.
Master (new)	M4NEW.❹
Coordinated	M4CORDn (n=1–9).
Transaction	M4TRAN.❹
Alternate Report Files	M4REPn (n=2 to 9).
Subfiles	M4SUBFn (n=0–9).
Deleted Master File Records	M4AUDIT.❹
Rejected Transaction	M4REJCT.❹

- If a file definition is generated for this subfile, another subfile must be generated that contains the file definition statement images.
- For M4OLD, M4CORDn, and M4SUBFn entries for IMS files, the JCL should specify the IMS DTF/ddname defined in the DBD.
- M4CORDn, M4SUBFn, M4OLD, M4NEW,❹ M4TRAN,❹ M4AUDIT,❹ or M4REJCT❹ can be specified for user I/O or VSAM password protect.

Working storage

Specifies the physical position of the work areas defined by the related file definition. Enter M4WORKnn (nn = 01–99). Numbers need not be consecutive.

Linkage section

Indicates the physical position of the address in the input parameter list of the area to be referenced by the specified linkage section file definition. Enter M4LINKnn (nn=01-99). M4LINKnn numbers need not be consecutive.

- If several parameters are passed, you can define a particular one by specifying only that DTF/ddname. This provides run time flexibility in the selection and ordering of the parameter list.
- Although an M4WORKnn or M4LINKnn entry is specified, no actual data file is being identified and, therefore, no JCL statements are needed. If the ddname list is being used and there are entries in it for M4WORKnn, they will be ignored. (Not available to CMS systems.)

Array processing

Type of Processing	Entry
Internal arrays	Blank.
Inputting array data from coordinated file	M4CORDn (n=1–9).
Array output to subfile	M4SUBFn (n=2 to 9).

Fields to use as key fields for input files (28-51)

Key1 (28-35)
Key2 (36-43)
Key3 (44-51)

Non-DL/I files

Specifies the name of the alternate keys. This applies to all input files other than the transaction files.

Entry	Result
Blank	File is processed using the record keys defined in the file definition for the file.
Any field name	File is processed using only the specified fields as keys. The file must be sequenced by these special keys as the level one segment of a record.
Field name of generic key	A request-read file is processed using this generic key unless overridden by an entry in operand B of the PR statement. This key must begin in position 1 of the record key and its length must be less than or equal to the length of the record key. A G must be specified in position [53].

- If the master and coordinated file record keys have the same numeric value, they will coordinate regardless of a difference in properties (type and scale).

  If both keys are character strings of different lengths, the master field is extended with blanks before comparison.

  If the key fields of the master file are changed, no transaction file is allowed in the run.

- If an R is specified in position [53], this entry must be blank.

ICF files

Field	Entry
1	Must be the name of the field controlling the coordinating (or direct access reading) process. ■ The contents of this field determine how a record from the coordinated file is used if it is not already in storage. ■ This field can be in any other file or any temporary field. ■ If the fields have different properties (type, length, or scale), the coordinating field is converted to the properties of the ICF record key. ■ An illegal conversion is treated as a "miss" or uncoordinated condition.
2	Blank.
3	Blank.

- These entries are not checked or used for IMS files when sequential coordination is needed. Only field 1 can be used if the IMS file is to be coordinated as an index coordinated file.
- For a coordinated relational file whose root segment is a multi-column keyed segment, this field must be defined as character and it must be overdefined by or composed of the host variables on the related user supplied WH statements. Refer to the section on multi-column keyed segments in the *VISION:Builder Reference Guide* for further information.

ICF (52)

Identifies the file in which the coordinating field (field 1) resides. It is available for IMS, relational, key sequenced VSAM, or ISAM VSE files only. The following qualifiers are permitted:

Code	Location of Field
N	New master file.
O or 0	Old master file.
1–9	Coordinated files 1–9.
T	Temporary field.
V	Linkage section.
W	Working storage.
X	Transaction file.❹

The entry must not be the qualifier of the coordinated file being specified (for example, if the coordinated file is M4CORD2, the qualifier entered here must not be 2). Note that if the ICF qualifier is blank, the file will be coordinated as a sequential file in the usual manner.

CORD (53)

Specifies that a generic key is used as the search argument, identifies the match file, specifies the type of coordination performed, or specifies the statement image subfile for file definitions.

If an entry is made in this specification for any coordinated file in the run, an entry must be made for every coordinated file in the run except for those containing an entry in the ICF qualifier.

Coordinated files

Entry	Result
Blank	Not valid for chained or unmatched coordination.
G	The search argument (specified in key1 [28-35]) used for RE/RG operators is the generic key. This key can be overridden in operand B of the PR statement.
M	The file is chain coordinated with the new master file work area. (A new master file need not be specified on the RC statement.)
1–9	The file is chain coordinated with the coordinated file specified.
U	Unmatched coordinated file records are made available for processing. The match file is the master file.
R	The file is moved and coordinated by you, using the RD, RE, or RG operators on PR statements. Request-read coordinated files can be used in combination with any type of coordination.

Note: If nonstandard coordination (M, U, 1–9) is specified for any file, the type of coordination must be specified for all files except those with a corresponding entry in the ICF specification.

Subfiles

Entry	Result
C	The subfile contains the statement images from automatic file definition generation. The user file name cannot be blank and the name cannot appear in En statements.
D	All dynamically deleted segments will be removed from the new master buffer before writing out the subfile record. This will only occur if the subfile specification in the corresponding En statement for entire record selection [43] is equal to N.

MOSAIC (54)

Specifies when to use memory optimized or MOSAIC processing for relational, mapped, and IMS files. Enter one of the following codes:

Entry	Result
Blank	Files are processed with the entire logical record in main storage.
M	MOSAIC processing is used in processing records for this file. At most, only one occurrence of each segment type is retained in main storage at any one time. MOSAIC processing can be specified for M4OLD and M4CORDn. When M4OLD is specified, M4NEW is not allowed in the same run.

- With MOSAIC processing, no distinction is made among the master file qualifiers (blank, N, or O) because all qualifiers refer to the same copy of the segment. Variable length field types (V) are not allowed in file definitions used with files being processed using MOSAIC.
- An entry of M is required for a relational master file that is update-in-place.
- When using FF or FL operators, MOSAIC processing can only be used with DL/I files. You cannot use FF and FL operators with relational or mapped files.

User I/O module name (55-62)

Specifies the name of the user written I/O module. Enter the link edited module name of the user written I/O routine. In addition to this entry, the following RF statement entries are also required if user I/O is specified: logical file name, DTF/ddname, and label.

- If using User I/O, you must reposition your own files after each IMS checkpoint call when using DBI (Data Base Interface).
- User I/O is not allowed on report summary files.

Array (63)

Unique identifier for an array.

Entry	Location of Data
A, B, E, G, H, J, K, M, Q	A specific array.
1–9	Coordinated file 1–9.

- This file is the same as the one specified in the DTF/ddname entry [20-27].
- Internal array qualifiers can be any normal alpha array qualifier or 1–9 as long as it has not already been assigned to a coordinated file. External array qualifiers must match M4CORDn file number (n).

Password/authorization ID (65-72)

Permits access to a protected data set. For VSAM files, enter any 8 characters (except all blanks) that is the password to access the file. The password is not printed in the VISION:Builder source listing but is replaced with the word PASSWORD.

For relational tables, enter the authorization ID of the creator or owner of the table. The authorization ID is not printed in the source listing, but is replaced by the word PASSWORD.

The authorization ID can also be specified on the LS statement. The order of precedence is:

- The entry on the RF statement, if provided.
- The entry on the LS statement, if provided.

If an authorization ID is provided, VISION:Builder supplies DB2 with a prefixed table name in the form of:

```
AUTHORIZATIONID.TABLENAME.
```

If an authorization ID is not provided, VISION:Builder provides DB2 with an unprefixed table name. In this case, DB2 uses the current SQL authorization ID. This comes from the SQLID parameter provided on the RP statement in the application.

If an RP statement is not provided, DB2 uses the signon user ID as the table prefix.

- Under OS/390, this is entered on the JOB statement.
- Under VSE and CMS, this is entered on the RP statement.

RG Statement - Run Control Group❹

The run group statement (RG statement) specifies which transaction groups are applied in a particular run. The default is all.

Position	Entry
01-08	Run name
09-10	Statement type
11-50	Transaction group
11-18	Group1
19-26	Group2
27-34	Group3
35-42	Group4
43-50	Group5

Run name (01-08)

Required entry

Identifies this statement as belonging to a particular run group.

Enter the name as entered on the RC statement run name.

Statement type (09-10)

Required entry

Identifies a transaction group statement. Enter RG.

Transaction group (11-50)

Required entry

Group1 (11-18)
Group2 (19-26)
Group3 (27-34)
Group4 (35-42)
Group5 (43-50)

Specifies the transaction groups used in the run. See general Rule A. These names must match the transaction definition group names specified for the master file used in the run.

Note: The maximum number of transaction group names per RG statement is eight; however, you can have as many RG statements in a run as needed.

Rn Statement - Output Specification

The Rn statement formats output and specifies the data for reports, alternate report files, alternate M4LIST files, report summary files, and/or subfiles.

Position	Entry
01-08	Request name
09-10	Statement type
11-13	Sequence number
14-15	Number of spaces before column
16	Qualifier
17-24	Field name
25	End line?
26	Non-print
27-28	Sort
27	Sequence
28	Descending?
29	Control
30	Subtitle
31-36	Summary entries
31	Total
32	Cumulative
33	Count
34	Maximum
35	Minimum
36	Average

Position	Entry
37	Delimited data output modifier
38-40	Extended output edit identifier
41-52	Summary entries
41	Percent/ratio or delimited data field decimal places
42	Percent/ratio field qualifier
43-50	Rn percent/ratio field (denominator)
51	Ratio
52	Percent
53-67	Output edit
68-71	Partial field entries
68-69	Starting character
70-71	Number of characters

If the Rn statement is used in a report that produces a graph, see to Chapter 5, *Graphic Fixed Format Statement Listing* for the Rn statement specifications for graphics output.

Request name (01-08)

Required entry

Identifies this statement and others accompanying it (if any) as one request. See general Rule A. Enter the same request name for all statements in the request.

Statement type (09-10)

Required entry

Identifies this as a reporting statement. Enter an R followed by a number (n) from 1–9. Up to nine reports can be generated per request. The number must be the same on all En, Fn, Jn, Kn, Rn, Pn, and Tn statements in a graph report statement group.

Sequence number (11-13)

Specifies the left to right sequence of the data on the report or subfile. See general Rule I.

Entry	Order of Printing Across the Page
Blank	Order of input.
Any 3 alpha, numeric, or alphanumeric characters	Ascending sequence.

Number of spaces before column (14-15)

Specifies the number of blank spaces to the left of a data column on the report.

Entry	Result
Blank	Two spaces before column (default).
0	No spaces before column.
1–99	1–99 spaces before column.

- This specification does not apply to the subfiles created.
- If a summary overflows the length of the field, the spaces before column provides room for printing the summary (if it fits).
- In formatted reporting and formatted sectional reporting, the nonprint entry used with a field in a title or summary line produces a default of zero spaces. Otherwise, spaces before the column are included in the length of the field when it is printed using an Fn statement.

Qualifier (16)

Identifies the type of field or file where the field exists.

Entry	Result
Blank or N	New master file field.
O or 0	Old master file field.
1–9	Coordinated files 1–9 field.
T	Temporary field.
F	Flag field.
X	Transaction file field.[4]

Entry	Result
W	Working storage.
V	Linkage section.
A, B, E, G, H, J, K, M, Q, 1 to 9	Array field (must match qualifier listed on the RF statement).

Field name (17-24)

Identifies the field to be output on a report or alternate M4LIST file or written to a subfile. Enter the name exactly as it appears on its file definition. See general Rule A.

- For relational subfiles, the column name will be this name unless the field is from a file definition that has an entry in [44-61] of the L0 statement. In this case, the name in [44-61] will be used.

  This also applies to any automatic file definition generated for the relational subfile. For the generated L0 statement, the entry in [11-18] will be the name on the Rn statement. The entry in [44-61] will be the column name, as discussed above.

- To output an array field, enter the field name as it appears on the array definition.

End line? (25)

Allows printing of data from each report record on more than one line of the report.

Entry	Result
Blank or N	Contents of next field print on the same line.
Y	Contents of next field print on the following line.

Non-print (26)

Suppresses printing of field values in a report.

Entry	Result
Blank or N	Field contents are printed.
Y	Printing of field contents is suppressed.

- If nonprint fields are used as sort or control breaks, the fields controlling the sort are not printed but the fields associated with them are printed with control break effects.
- Nonprint fields can be used with subtitle break fields to control spacing or page ejection without printing.
  - If summaries are specified, space is allowed for the field as if it were a normal column. However, detail data is suppressed and only summary values are printed.
  - If summaries are not specified, no column is allowed for this field and an entry in number of spaces before column is ignored.

  When a Y is entered, a blank number of spaces before column specification is interpreted as zero. This field also indicates embedded fields in title lines of formatted reporting output that do not appear in detail line output.

Sort (27-28)

Sequences data in a different order than the one in which it is stored in the database. You can define major to minor sort sequence and ascending or descending order for output.

Sequence (27)

Signifies the order in which the data is sorted for a report. The entry must be blank for type V fields.

Entry	Order by Which Fields Will be Printed on the Report
Blank	Order of input file.
1–9	1–9 (major to minor) sequence.

Entering the same number in more than one field is not valid; a diagnostic message results.

The sort sequence does not alter the order of the columns across the page.

Subfile Considerations

Key	The sort sequence designates segment key fields when generating an automatic subfile file definition. Entering a 1 with IMS, VSAM, ISAM, causes that field to be the key of the subfile. If this entry is blank on all Rn statements, the field specified on the first Rn statement is the key of the subfile.
Relational Not Null	For relational subfiles, fields specified with sequence are created with the Not Null specification.

Descending? (28)

Sequences numbers from the largest to the smallest.

Entry	Resulting Sort
Blank or N	Ascending order.
Y	Descending order.

- For character string fields, ascending sort means A–Z and descending Z to A.
- For relational subfiles with an accompanying automatic file definition, specifying Y causes the generated definition LS statement to have a D (descending) in segment order.

  If left blank, the specification will be blank (unordered).

Control (29)

Specifies the field as a control break field for fields that are not type V. Specifies how a type V field will be split between report pages or output to a subfile.

Entry	Result
1–9	Not valid for type V fields. Enter numbers from 1–9 opposite the fields used as control break fields. The field value prints once when each control break occurs. All other occurrences of that value are suppressed. The number 1 is the most major level control break and 9 the most minor. A change in value of this field causes control breaks to occur on the fields at that, and more minor, levels. Multiple fields can be selected as control breaks at any level; a change in value of any of them causes a control break to occur. Control breaks are usually used with summaries but can be used solely to suppress printing. Summaries are not allowed on any fields specified as control breaks. At level 1, a blank line is output when a control break occurs. When a control break occurs, requested summaries print on the report. A record containing control break values and summaries can be output to the report summary file if requested by the user.
Blank	All occurrences of the field are printed. A type V field automatically folds to the next line within the column width. The fold occurs between words. Any leading blanks are removed from the beginning of the next line. Printing of type V fields begins on the following page if the entire field will not fit on the remaining lines of the current page. Subfile output of type V field is identical to the internal VISION:Builder format
S	Same as the blank entry, except that a type V field begins printing on the current page and is split between report pages if the bottom of the current page is reached before the entire type V field is printed.

Note: The Blank and S entries cause breaks in type V fields between words and left-alignment of the folded data by removing all blanks between words at the break point.

Entry	Result
H	A type V field folds into multiple report lines in units equal to the column width without regard to word boundaries. Blanks are not removed; all blanks are preserved. Printing of type V fields begins on the following page if the entire field does not fit on the remaining lines of the current page.
B	Same as the H entry, except that a type V field begins printing on the current page and is split between report pages if the bottom of the current page is reached before the entire type V field is printed.
Note: The H and B entries cause breaks in the type V field at units equal to the column width. No alignment occurs because no blanks are removed.	
X	The type V field is output as a PL/I-compatible varchar field; that is, field is output with 2-byte prefix containing the current length of the field followed by the significant characters in the field (represented by the current length) padded with blanks up to the maximum length of the field.
Y	The type V field is output as a long character field containing the significant characters in the field (represented by the current length) padded with blanks up to the maximum length of the field.

Subtitle (30)

Prints a subtitle when the value of the field changes. The field must also be specified as a control break field by entering 1–9 in control [29].

This entry must be blank for type V fields.

Entry	Result
Blank	No subtitles requested.
S	Subtitle requested. A change in value of the specified field causes the new value of this field to print as a subtitle after printing summary information related to the last value of the field. ■ At a control break, S-type subtitles are printed at the current control break level and more minor levels. ■ At a page break, S-type subtitles are printed in the same format as previously printed only if you have specified repeating subtitles. (See M4PARAMS in the *VISION:Builder for OS/390 Installation Guide* or the *VISION:Builder for VSE Installation Guide* or En statement position [61].) ■ At a page break, S-type subtitles print if you have specified repeating subtitles.

Entry	Result
P	Page subtitles requested. P signifies that a field value change starts a new page after printing summary information. The new field value prints as a subtitle on all subsequent pages below the page title but above the column headings.

- Fields designated with P or S subtitles cannot be summarized nor are column headings printed for them. Summaries are taken on columnar data only.
- For both P-type and S-type subtitles, if the same control break level is designated for two or more fields, the values print on one line of the report in the order of sequence numbers [11-13] assigned to each within the control break level. All fields at the most major level print first, followed by all fields at the next minor level, and so on.
- If printing of an S-type subtitle field is suppressed because of a control break, the most major level field not suppressed prints starting at the left of the field.
- P and S are not applicable for formatted reporting.

Summary entries (31-36) and (41-52)

The following summary entries (total, cumulative, count, maximum, minimum, average, percent, ratio) are used to indicate the type of summary to be computed and at which control break level the summary will be produced.

A control break occurs each time the values in the specified control break fields change. Control break fields are specified using the Rn statement control [29] entry. Control break significance is indicated by level assignments from 1–9, with 1 being the most major and 9 being the most minor.

When a control break occurs at a specified level, summaries are computed and summary lines are printed for all the more minor control break levels specified and for the level at which the control break occurred. All the valid field values processed since the last control break occurred at each level are used to compute the summaries, even if printing is suppressed by a lower level control break.

Summary lines are printed with the same vertical spacing as detail lines; however, an extra blank line is placed before, after, and between each level of summaries.

Summary labels

When summaries are specified for a report, 14 spaces are automatically reserved on the left side of the entire report page for summary labels. Space for the report data columns is assigned following the 14 spaces reserved for summary labels.

- Each summary print line contains a summary label and the specified summary information.
- The summary label contains the name of the control break field for that control break level along with the type of the summary being printed. One summary line is printed for each summary type specified at the control break level. Note that because the control break field name is used in the summary label, meaningful field names should be chosen as control break fields.
- The printing of summary labels can be controlled by using the En statement labels on summary lines [40] entry. If several fields are specified as control break fields for the same level, the first field name is used in the summary label. You can designate which field name to use in the summary label by using the "L" specification in the total entry [31] of the Rn statement that specifies the control break field [29] for the report.

Grand summaries

Grand summaries can be automatically or selectively produced at the end of the report. There is an M4PARAMS installation option to specify YES or NO as to whether automatic grand summaries should be printed.

- The default setting is NO.
- When the option is set to YES, all summaries specified for the report will automatically have grand summaries computed and printed on a separate page at the end of the report.
- The M4PARAMS setting can be overridden for all reports in the application by using the RP statement AUTOGRND parameter.

  The setting can be overridden for each individual report in the run by using the En statement autogrand [59] entry.

Summary limitations

Summary entries are not allowed for the following:

- Any field on an Rn statement that also specifies the field as a control break field (1–9 in control [29]). Two Rn statements would be needed: one for control and one for summaries.
- Variable length (type V) fields.
- Character fields whose length exceeds 15 positions, the summaries for total, cumulative, average, percent, or ratios are not permitted.

Field handling

The contents of the detail data fields used for summaries are inspected to determine how they will be handled. Fields that are detected as invalid (*) or nonexistent (-) will not be included in the computed summaries.

Detail data fields are also scanned to determine if they are empty (that is, the contents are all blanks in character fields or the value is zero in numeric fields). Normally, a character field that is empty is not included in summaries, while a numeric field that is empty is included in summaries. This can be controlled by using the En statement empty field control [58] entry.

Plus signs (+) as summary output values

When a plus sign (+) prints for the computed summary value, it indicates that there is a problem with the computed summary value. The problem is one of the following situations:

- The edited summary value contains more characters than the report data field's column width can accommodate (which includes the spaces before column value).
- The percent or ratio field total was zero causing a division by zero.
- All the values for a maximum, minimum, average, percent, or ratio summary were empty (zero or blank). Because there are no values to use in the computation of these summaries, the plus sign indicates an incomplete summary.

Report summary files

When a report summary file is specified for a report, a record is written out when each control break occurs. The record consists of each requested summary value and the corresponding control break field values for the level at which the control break occurred. If grand summaries are produced, autogrand is YES and/or G level summaries are specified, a record is written out at the end of the report containing the grand summary values.

Total (31)

Indicates that a total will be produced for the specified field whenever a change in the value of the corresponding control break fields occurs at the specified control break level or a higher control break level (numerically lower). A total represents the accumulation of field values since the last control break occurrence at a control break level.

Entry	Result
Blank	No total is produced.
1–9	The control break level at which a total is produced.
G	Only produce a grand total at the end of the report.
L	Specifies that this field name is used for the summary label. ■ This entry must be specified on the Rn statement that also contains the control break [29] entry specification. ■ The summary label will include this field name on all summary lines at the corresponding control break level. ■ If the L is specified on more than one Rn statement for the same control break level, the field name from the last Rn will be used for the summary label. ■ If this specification is not used, the first field name with a control break [29] entry is used in the summary label.

Cumulative (32)

Indicates that a cumulative total will be produced for the specified field whenever a change in the value of the corresponding control break fields occurs at the specified control break level or a higher control break level (numerically lower). A cumulative total represents an accumulation of field values since the beginning of the report.

Entry	Result
Blank	No cumulative total is produced.
1–9	The control break level at which a cumulative total is produced.
G	Only produce a grand cumulative total at the end of the report.

Count (33)

Indicates that a count will be produced for the specified field whenever a change in the value of the corresponding control break fields occurs at the specified control break level or a higher control break level (numerically lower). A count represents the number of occurrences of the field since the last control break occurrence at a control break level.

Entry	Result
Blank	No count is produced.
1–9	The control break level at which a count is produced.
G	Only produce a grand count at the end of the report.

Maximum (34)

Indicates that a maximum will be produced for the specified field whenever a change in the value of the corresponding control break fields occurs at the specified control break level or a higher control break level (numerically lower). A maximum represents the maximum value encountered in the field since the last control break occurrence at a control break level.

Entry	Result
Blank	No maximum is produced.
1–9	The control break level at which a maximum is produced.
G	Only produce a maximum at the end of the report.

Minimum (35)

Indicates that a minimum will be produced for the specified field whenever a change in the value of the corresponding control break fields occurs at the specified control break level or a higher control break level (numerically lower). A minimum represents the minimum value encountered in the field since the last control break occurrence at a control break level.

Entry	Result
Blank	No minimum is produced.
1–9	The control break level at which a minimum is produced.
G	Only produce a grand minimum at the end of the report.

Average (36)

Indicates that an average will be produced for the specified field whenever a change in the value of the corresponding control break fields occurs at the specified control break level or a higher control break level (numerically lower). An average represents the average value encountered in the field since the last control break occurrence at a control break level.

The average equals the total of valid values in the detail lines (since the last average on this control break level) divided by the number of entries included in this total. The output format of the average summary is the same as the detail field format. For example, if two integers are averaged, the result has no decimal places.

Entry	Result
Blank	No average is produced.
1–9	The control break level at which an average is produced.
G	Only produce a grand average at the end of the report.

Delimited data output modifier (37)

Modifies the default output format for delimited data (CSV or tab-delimited output method) fields.

Entry	Result
Q	Always enclose the character field within quotes (“) regardless of whether it contains an imbedded comma (,).
T	Truncate trailing zeros to the right of the decimal point of the numeric field up to either the decimal point (default) or the number of decimal places specified in the Percent/Ratio or Delimited Data field decimal places (column 41) entry. Only valid for CSV or tab-delimited output of numeric type fields.

Extended output edit identifier (38-40)

The 3-position character value is used to link to a specific Qn statement that contains an extended output edit pattern.

Percent/ratio or delimited data field decimal places (41)

Specifies the number of decimal places to print in the result of the percent/ratio computation or CSV output of a numeric field.

Entry	Result
Blank	Percent or Ratio field: ■ Ratio - 3 decimal places in the result. ■ Percent - 2 decimal places in the result. CSV Output field with T in Modifier entry (column 37): ■ Truncate trailing zeros up to, but not including, the decimal point.
0–9	Percent or Ratio field: ■ 0–9 decimal places in the result. CSV Output field with T in Modifier entry (column 37): ■ Truncate trailing zeros up to the number of decimal places specified.

Percent/ratio field qualifier (42)

Indicates the location of the field to be computed with a percent/ratio.

Entry	Location of Field Name A
Blank or N	New master file field.
O or 0	Old master file field.
1–9	Coordinated files 1–9 field.
T	Temporary field.
X	Transaction file field.[4]
W	Working storage.
V	Linkage section.
A, B, E, G, H, J, K, M, Q, 1 to 9	Array (qualifier must match the one listed on the RF statement).

Rn percent/ratio field (denominator) (43-50)

Names the field on which the percentage or ratio is produced.

Entry	Result
Blank	No percent/ratio is produced.
Field name	Percent/ratio is produced with this field.

- The field must also appear on an Rn statement for the same request. It can be a nonprint field.
- A total is computed (even if not specified) for use in the percent/ratio computation.
- The percent/ratio is computed as follows:

$$\frac{\text{Field Total}}{\text{Percent/Ratio Field Total}} = \text{Result}$$

  The result prints under the field name column on the report and the summary label contains the percent/ratio field name, followed by the designation PCT or RATIO.

Ratio (51)

Indicates that a ratio will be produced for the specified field whenever a change in the value of the corresponding control break fields occurs at the specified control break level or a higher control break level (numerically lower). A ratio represents the accumulation of field values since the last control break occurrence at a control break level.

This entry must be blank for type V fields.

Do not use this entry with subfiles.

Entry	Result
Blank	No ratio is produced.
1–9	The control break level at which the ratio is produced.
G	Only produce a grand ratio at the end of the report.

Note: Three decimal places print in the result unless overridden by an entry in percent/ratio field decimal places [41].

Percent (52)

Indicates that a percent will be produced for the specified field whenever a change in the value of the corresponding control break fields occurs at the specified control break level or a higher control break level (numerically lower). A percent represents the accumulation of field values since the last control break occurrence at a control break level.

This entry must be blank for type V fields.

Do not use this entry with subfiles.

Entry	Result
Blank	No percent is produced.
1–9	The control break level at which the percent is produced.
G	Only produce a grand percent at the end of the report.

Note: Two decimal places print in the result unless overridden by an entry in percent/ratio field decimal places [41].

Output edit (53-67)

Overrides the output edit entries on the L0, TF, or B0 statements. This edit pattern can consist of either of two styles: the delimiter style or the COBOL style. The delimiter style pattern can be used when editing character data or numeric data. The COBOL style pattern can only be used with numeric data. Do not use this entry with numeric field type E.

Requirements for output editing of character string data, numeric data, and variable length fields follow.

Character string data

The edit pattern must be the delimiter style pattern and must use the system delimiter character as the select character. Any other EBCDIC characters, including embedded and leading blanks, are valid. The edit pattern is entered as a picture with the delimiter (#) being used as a select character. All other characters are edit insertion characters (that is, a date pattern can be specified as ##/##/##).

The first character in the output edit pattern entry always represents the leftmost character. Truncate characters on the right by not entering delimiters to represent them. If left truncation or extraction of characters in the middle of a field is preferred, use partial fielding [68-71] to specify the partial field start position and number of characters to be printed.

Numeric data (P, Z, and F only)

The edit pattern enables the user to truncate digits following the decimal point or nonsignificant zeroes to the left of the decimal point. The pattern causes digits surrounding the decimal point to print.

The period (.) has a special meaning in the override edit pattern. The first period is interpreted as a printable decimal point and is used to align decimal points. If no period exists, only the integer portion of a number is printed. Leading zeroes are suppressed if they are the leading characters of the picture. When a negative value is output, the sign is printed to the left of the first digit or replaces the first leading zero.

The edit pattern can use the system delimiter pattern as described for character string data.

Additionally, for numeric data, a COBOL style edit pattern can be specified. See the table on page 2-196 and rules for output edit for details on how to specify the edit pattern.

Date fields (type D)

The edit pattern must be a picture representing the desired output format for the date, for example MM/DD/YY. Refer to the *IBM Language Environment Reference Manual* for a list of valid picture characters.The picture is used as a parameter in a call to CEEDATE.

Variable length fields (type V fields)

Enter the output edit length (column width) within which the type V field prints. The value can be from 1–the report page width minus spaces before column. It overrides the output edit length specification in the L0 or TF statements. The value can be entered anywhere in these columns.

- If output edit is used for time data, the hours, minutes, and seconds have no significance. The number is treated the same as a packed decimal (P) number with the specified number of decimal places.
- If output edit is specified for a field that is summarized with a count, percent, or ratio, the field will be edited on the detail line, but the summary (count, percent, ratio) will not be edited. Total, cumulative, and average summaries carry the editing characteristics.

Symbol	Result
Digit select (9)	Represents a character position that contains a numeral from the field name specified in [17-24]. Valid for numeric fields only.
Digit select (Z)	Same as above, except when the result contains a zero as a leading character, the zero is replaced by a space (blank). Signs do not print.
Check protection (*)	Same results as Z, except a zero in a leading position is replaced by the asterisk (*).
Decimal position (.)	Represents a decimal point for alignment in the edit picture. For the output, the decimal point represents a position into which a character is inserted.
Grouping character (,)	Represents an output position into which a character is placed.
Sign control (±)	Used as editing sign control symbols to position the edit sign.
Currency symbol ($)	Represents placement of the currency symbol.
System delimiter (#)	Character or numeric digit select.

Note: All symbols can be varied by M4PARAMS entries as long as each symbol chosen is unique among all the editing symbols and the system delimiter. No editing symbols can have a value in the range of hexadecimal X'00' through X'30'.

Rules for output edit

- Source digits can be edited to the output positions with or without suppression of leading zeroes or check protection (for example, 99.9=no suppression, ZZ.9=suppress leading zeroes to decimal point, **.*=check protects the entire field).
- A leading or trailing sign can be specified (for example, +99, -99, 99-, 99+). Sign control symbols produce the following results:

Symbol in Edit Picture	Result if Positive Data Item (Includes Zero)	Result if Negative Data Item
+	+	-
-	SPACE	-

- A leading sign can be specified as fixed (leftmost position on output) or floating to the left of the first significant output character (for example, +99.9=fixed, +++.9=floating).
- A currency symbol can be specified and can be fixed (leftmost position on output) or can be floating (for example, $99.9=fixed, +$$$.9=floating). The currency symbol and the sign symbols are mutually exclusive as fixed insertion characters in a given picture.
- Floating insertion of leading sign or currency symbol is indicated by using a string of at least two occurrences of the same character ($+-) to represent the leftmost numeric positions in the output (for example, +99.9=fixed, +++.9=floating). The currency symbol and the sign symbols are mutually exclusive as floating insertion characters in a given picture. A fixed insertion character can also be specified in the same picture with a floating insertion character as long as it is a different character.
- Floating insertion of sign or currency symbols implies suppressing leading zeroes. That is, they act as suppression symbols (for example, $$,$$$.99).
- Suppression symbols (Z$*+-) can occur in an edit picture as follows:
  - Any or all of the leading numeric positions to the left of the decimal point are represented by suppression symbols (for example, $$,$$$.99).
  - All of the numeric positions in the edit picture are represented by suppression symbols (for example, $$,$$$.$$).

  If the suppression symbols appear only to the left of the decimal point, any leading zero in the data that appears in a character position corresponding to a suppression symbol in the picture is replaced by the space character or check protection character. Suppression ends at the first nonzero digit in the data or at the decimal point, whichever comes first (for example, $$$.99 with data 001.23 produces $1.23).

If all numeric positions in the picture are represented by suppression symbols and the value of the data is not zero, the result is the same as if the suppression characters were only to the left of the decimal point.

If the value of the data is zero, the entire output will be spaces if all numeric positions in the picture are represented by suppression symbols (for example, $$.$ with data 00.0 produces all spaces) or, if check protection is used, all positions will be the protection character, except a decimal character will remain (for example, **.* with 00.0 produces **.*).

Any grouping characters embedded in the string of suppression characters or to the immediate right of the string will be suppressed if the left adjacent numeric position is suppressed (for example, $,$$$.99+ with data 0012.34 produces $12.34+).

- Using a suppression symbol throughout a field will also cause suppression of a trailing sign on an all zero field (for example, ZZ.ZZ+ with value +00.01 produces .01+, while if used with value +00.00, produces all spaces). Delimiters should not be used with Z picture edit (for example, ZZZ,ZZZ.## is illegal; ZZ,ZZZ.99 is valid).
- The grouping character can occur on either side of the decimal position character (for example, $$,$$$.999,99 international metric standard). It does not need to occur every three digits (for example, $$$$,$$$.99).
- Truncating nonsignificant digits (leading and/or trailing zeroes) is valid.

  Truncating significant trailing digits is valid (for example, 99.99 with data 12.345 produces 12.34).

  Truncating significant leading digits is not valid (for example, 9.99 with data 12.34 produces + or the uneditable symbol).
- A trailing character, if specified, cannot be the same symbol used as a floating or leading character in a given pattern (for example, +99.9+ is illegal; +99.9- is valid).

Note: The M4PARAMS settings for the editing symbols at request processing time are used to decode and validate the edit patterns. If an M4PARAMS with changed editing symbols is used at reporting time, the edited output field will use the changed symbols.

Partial field entries (68-71)

Partial field entries (starting character, number of characters) allow processing of any part of a character string field.

Both starting characters [68-69] and number of characters [70-71] must be specified. Starting and number of characters must fall within the range of the operand specified.

Note: Partial field flags cannot be used on the Rn statement.

Starting character (68-69)

Location of first character used in the partial field.

Entry	Result
Blank	No partial fielding.
1–99	Position of character beginning the partial field.

Number of characters (70-71)

Number of characters forming the partial field.

Entry	Result
Blank	No partial fielding.
1–99	Number of characters that form the partial field (up to the maximum length of the field).

RP Statement - Run Control Group

The run parameters statement (RP statement) supplies miscellaneous parameters. These parameters are effective only in the run to which they apply.

Position	Entry
01-08	Run name
09-10	Statement type
11-18	Parameter name
19-71	Parameter value

Run name (01-08)

Required entry

Identifies this statement as belonging to a run group. Enter the same name as on the RC statement.

Statement type (09-10)

Required entry

Identifies a run parameter statement. Enter RP.

Parameter name (11-18)

Enter one of the parameter names listed in the table on page 2-200.

Parameter value (19-71)

Required entry

Specifies the value assigned to the parameter named in [11-18]. Enter one of the values shown in the following table.

Parameter Name	Acceptable Parameter Values and Action Taken
#DUMP	None. *Action:* Abends the job stream and provides a dump for a run with terminal errors and a user ABEND code of 0010. OS/390 users must supply a SYSUDUMP DD statement. VSE users must include an OPTION PARTDUMP statement. # denotes the system delimiter. A type 9 error provides a dump even if RP#DUMP is not specified.

Parameter Name	Acceptable Parameter Values and Action Taken
31-BIT	YES *Action:* Allocates storage for table-lookup tables, arrays, and file buffers above the 16-MB line whenever possible. For VSE, a value of YES is ignored.
	NO *Action:* Suppresses allocation of storage for table-lookup tables, arrays, and file buffers above the 16-MB line, and processes in 24-bit mode. For VSE, a value of NO is always assumed.
ABEND OS/390 only	YES Position [19-21] *Action:* Terminates the job step if a definition or processing run return code is 4 or 16 in the job stream. Ignore this specification for scan-only or sample report runs. **Note:** The values in M4PARAMS for CONDCOD2 and CONDCOD4 (delivered as 4 and 16) are actually used to compare against the job step return code.
	SYSTEM Position [19-24] *Action:* Does not invoke the VISION:Builder I/O synad error routine when an I/O error occurs; instead, returns control back to the operating system for normal I/O error ABEND.
AUTOGRND	YES *Action:* Generates automatic grand summaries. This parameter overrides the grand summary setting in M4PARAMS.
	NO *Action:* Does not generate automatic grand summaries. This parameter overrides the grand summary setting in M4PARAMS.

Parameter Name	Acceptable Parameter Values and Action Taken
CHKORDER	Name of VISION:Builder input file to be checked: M4OLD, M4TRAN[4], M4CORDn. CHKORDER must be blank for array files. *Action:* Checks the order of incoming segments according to the segment key specification in the associated file definition. Does not check records that fall outside the range of the start search, end search entries on the RC statement for segment order. Does not check files that are using memory optimized (MOSAIC) processing. Performance degradation may result while using this parameter.
CORDONLY	None. *Action:* Bypasses the follow-up pass. Bypasses execution of requests when the ECORD flag is equal to all Xs, whether or not coordination has taken place.
DB2[D]	Enter the DB2 SSID. Position [19-26] The installation-dependent plan name. Position [27-34] *Action:* Identifies that CALL ATTACH is being used to access DB2 tables.
DECMSGS	YES *Action:* Displays Decode and Compile phase information and warning messages.
	NO *Action:* Suppresses Decode and Compile phase information and warning messages.
EXPLAIN[D]	Any 9 digits as the query number. Default is zero. Position [19-27] *Action:* Enables DB2 to write performance information out to AUTHORIZATIONID.PLAN_TABLE for each EXPLAINable SQL statement and uses the query number as a key. The query number is incremented by one after each EXPLAINable SQL statement. The user is responsible for listing EXPLAIN information from AUTHORIZATIONID.PLAN_TABLE.

<table>
<tr><th>Parameter Name</th><th>Acceptable Parameter Values and Action Taken</th></tr>
<tr><td>FREESIZE

(OS/390)

(VSE)</td><td>A value from 1K–1024K. The value of K is always 1024.

Leading zeroes can be omitted.

Action: Frees specified storage during the processing step of VISION:Builder. Releases the storage after files have been opened. Rounds up the specified value to a 2K multiple, or a 4K multiple under OS/390.</td></tr>
<tr><td>LISTCNTL</td><td>Nine characters consisting of Ys and Ns.
Position [19-27]

Action: Overrides the source statement listing specification [31] on the RC statement.

The parameter values correspond to the following source listing elements:
<table>
<tr><th>Element</th><th>Col #</th><th>Default</th></tr>
<tr><td>File Summary – only for OS/390</td><td>19</td><td>Y</td></tr>
<tr><td>Instream definition statements</td><td>20</td><td>Y</td></tr>
<tr><td>Instream definition glossaries</td><td>21</td><td>Y</td></tr>
<tr><td>Instream requests/procedures</td><td>22</td><td>Y</td></tr>
<tr><td>Cataloged requests/procedures</td><td>23</td><td>Y</td></tr>
<tr><td>Mapping requests/procedures G</td><td>24</td><td>N</td></tr>
<tr><td>SQL summary/statistics D</td><td>25</td><td>Y</td></tr>
<tr><td>MOSAIC summary/statistics I D G</td><td>26</td><td>Y</td></tr>
<tr><td>Generated requests (from ASL procs)</td><td>27</td><td>N</td></tr>
</table>
A value of Y causes the element to be listed, while N causes the element to be suppressed. A value of blank in any position allows the explicit specification on the RC statement to remain in effect for that element. If there is no explicit specification, blank is equivalent to the default.</td></tr>
</table>

Parameter Name	Acceptable Parameter Values and Action Taken
MAXGETMN	Value of the form nnnnK, where nnnn is a number from 1–8192 and K indicates that the total value is a multiple of 1024. Leading zeroes can be omitted from nnnn. The value is rounded up to a multiple of 4K for OS/390 and CMS systems. It is not used on VSE systems. *Action:* Overrides the upper GETMAIN limit with the specified MAXGETMN value, causing more or less storage to be obtained. The storage obtained is in addition to that required to hold VISION:Builder, system I/O, and so on. The default value is specified in M4PARAMS.
MULTILIB	OFF Position [19-21] *Action:* Inhibits the MULTIPLE M4LIBs process. VISION:Builder will not attempt to locate or use alternate libraries. Any JCL or ASSGN for alternate libraries will be ignored.
	M4LIBn Position [19-24] *Action:* Activates a limited alternate library process. VISION:Builder will only attempt to locate and use the alternate libraries specified. The value of n can be 1 through 9. JCL must be provided for libraries. Any JCL specifying libraries other than those on RP MULTILIB statements will be ignored. Multiple RP MULTILIB statements can be specified.
	M4LIB Position [19-23] *Action:* Same as M4LIBn. Specifies the default primary library (M4LIB) as an alternate library.
ONEBUFFR	None. *Action:* Specifies that M4OLD and M4NEW share one common area of storage for the master file record data. Parameter is not valid when a transaction file or update-in-place is specified.

Parameter Name	Acceptable Parameter Values and Action Taken
OVERRIDE	Fixed form as follows: *Action:* Permits overriding of VISION:Builder's standard DTF/ddnames and SYSnnn numbers. Position [19-25] The standard VISION:Builder DTF/ddname to which this override applies. Any user file except M4INPUT and M4LIST. Each override must be specified on a different RP statement. A nonexistent VISION:Builder DTF/ddname specification will result in an error. VSE Position [27-33] OS/390 Position [27-34] The new DTF/ddname must be valid for the operating system environment. For OS/390 only, #, @, and $ are considered valid characters in an override ddname. If only overriding SYSnnn, this field must contain a standard VISION:Builder DTF/ddname.
	Position [35-40] (VSE) The user SYSnnn specification to override the VISION:Builder DTF logical unit assignment. VISION:Builder uses the fixed set of logical unit assignments specified in MARKUNIT. This default SYSnnn number can be changed if the following rules are followed. nnn can contain 001 through 240, PCH, RDR, LST, IPT. Not required if the default SYSnnn assignment is used.
PROMSGS	YES *Action:* Displays File Processing phase information and warning messages.
	NO *Action:* Suppresses File Processing phase information and warning messages.

Parameter Name	Acceptable Parameter Values and Action Taken
REPTSIZE	Value of the form nnnnK, where nnnn is a number from 1–8192 and K indicates that the total value is a multiple of 1024. Leading zeroes can be omitted from nnnn. The value nnnn is rounded to a multiple of 2. *Action:* Applies to a single-step, no-sort run (where R has been specified in the RC statement resource optimization report file entry). Overrides the default main storage amount specified in M4PARAMS that VISION:Builder allocates when this optimization method is chosen.
REWIND UNLOAD NOREWIND (VSE only)	ALL, M4OLD, M4NEW, 4 M4TRAN, 4 M4SORT, M4CORDn, M4REPn, M4SUBFn, M4REPO, M4REPI, M4REJCT, 4 M4SSOUT, M4AUDIT 4 *Action:* Specifies named files to be rewound, unloaded, or not rewound in accordance with the following rules. Note that all error conditions will result in run termination. Only one RP statement with the ALL parameter can be entered per run. For a given DTF name, one RP statement specifying that name can be entered to override the ALL specification for a file or to specify tape positioning for a given file in the absence of an ALL statement. A label specification of X on an RC or RF statement overrides an RP statement ALL specification. This also causes an RP statement for a specific DTF name to be rejected with an error message. An RF statement for user I/O overrides RP statement specifications. An RP statement specifying the DTF name of a nonexistent file receives an error message. An RP statement is only valid for tape files.
RPO	None. *Action:* Compiles the report step. You cannot use the RPO option if you have requested a Report Summary file.

Parameter Name	Acceptable Parameter Values and Action Taken
RPTMSGS	YES *Action:* Displays Report Generation phase information and warning messages.
	NO *Action:* Suppresses Report Generation phase information and warning messages.
SORTSIZE	Value of the form nnnnK, where nnnn is a number from 1–8192 and K indicates that the total value is a multiple of 1024. Leading zeroes can be omitted from nnnn. VISION:Builder rounds the value nnnn up to a multiple of 2. *Action:* Applies to a single-step, sort run (where R has been specified in the RC statement resource optimization report file entry). This entry overrides the default SORT working storage amount specified in M4PARAMS that VISION:Builder passes to the SORT when the above-mentioned optimization technique is chosen.
SQL/DS (D)	Enter the user ID. Positions [19-26] Enter the password required to access SQL/DS. Positions [27-34] *Action:* Enables the user to use SQL/DS under VSE.
SQLID (D)	For OS/390 only. Enter the SQLID. Positions [19-26] *Action:* Enables the supplied SQLID to be used as the authorization ID by DB2 for any default table qualification and for checking required table privileges.

Parameter Name	Acceptable Parameter Values and Action Taken
SYSDATE	A 6-digit value that must be specified in the same format as that supplied for the TODAY flag in M4PARAMS; if the TODAY flag format in M4PARAMS is YYMMDD, the RP statement value must also be in the same format. *Action:* Sets the specified date into all date flags (DATE, ISDATE, JULANX, JULIAN, LILIAN, TODAY, and TODAYX) and uses the date as the report date when requested. This date has no effect on the system date shown on VISION:Builder listings or glossaries. When the year (YY) is specified as 00 through 89, the century will be interpreted as 20.
SYSDATE4	An 8-position field that supports 4-digit years in the format YYYYMMDD. It functions the same as the SYSDATE parameter, but has no dependency on the date format setting in M4PARAMS.

RT Statement - Run Control Group

The relational tables statement (RT statement) provides the information needed for each subfile in the run that is a relational table.

Position	Entry
01-08	Run name
09-10	Statement type
11-18	File name
27	Table control
30-37	Authorization ID
38-55	Table name
56-63	Database name or DBSPACE owner
64-71	TABLESPACE or DBSPACE

Run name (01-08)

Required entry

Identifies this statement as belonging to a particular run group. Enter the same name as entered on the RC statement.

Statement type (09-10)

Required entry

Identifies a relational table statement. Enter RT.

File name (11-18)

Required entry

Identifies the VISION:Builder logical file to which this statement applies. Enter one of the names M4SUBF0–M4SUBF9.

Table control (27)

Required entry

Entry	Result
C	Create - creates a table and inserts all entries into the created table. If the table already exists, an error message is issued.
D	Delete/Replace - deletes all existing rows from a table and inserts new rows. If the table does not exist, an error message is issued.
I	Insert - inserts all entries into the existing table. If the table does not exist, an error message is issued.

Entry	Result
R	Drop/Replace - drops the existing table, creates a new table, and inserts new rows. If the table does not exist, it is created and the rows are inserted. Caution: The action R causes a DROP that will lose all synonyms, authorizations, and so on, associated with the table.

Note: These actions can only take place if the application user has proper authorization provided to DB2 by the system administrator.

Authorization ID (30-37)

The authorization ID of the owner of the table. This specification takes precedence over any authorization ID specification on the RF statement.

- If an authorization ID is provided on either the RT or RF statement, VISION:Builder supplies DB2 with a prefixed table name in the form of:

  ```
  AUTHORIZATIONID.TABLENAME.
  ```

- If an authorization ID is not provided, VISION:Builder supplies DB2 with an unprefixed table name. In this case, DB2 uses the current SQL authorization ID. This comes from the SQLID parameter provided on the RP statement in the application.
- If the RP statement is not provided, DB2 uses the signon user ID as the table prefix. Under OS/390, this is entered on the JOB statement. Under VSE and CMS, this is entered on the RP statement.

Table name (38-55)

Required entry

Enter the name of the table as it is to be known to DB2.

Database name or DBSPACE owner (56-63)

For DB2, enter the database name. If left blank, the system default takes effect.

For SQL/DS, enter the name of the owner of the DBSPACE. If this entry is left blank, the system default takes effect. If an entry is made here, an entry must also be made for DBSPACE [64-71].

TABLESPACE or DBSPACE (64-71)

For DB2, this entry identifies the TABLESPACE name. If left blank, the system default takes effect.

For SQL/DS, this entry identifies the DBSPACE name. If left blank, the system default takes effect.

For TSO ATTACH and CALL ATTACH, if this entry is left blank and R is specified for table control [27], VISION:Builder calls DB2 to perform a COMMIT after the DROP, prior to the CREATE.

TB Statement - Table Definition

The table definition statement (TB statement) defines the attributes of the table to VISION:Builder. When you change the attributes of a table, you must recatalog the table.

Note: Displacement tables must not exceed a total of 32,000 bytes (the length of the result field multiplied by the number of entries). Binary and sequential tables have a maximum number of entries of 32,767.

Position	Entry
01-08	Table name
09-10	Statement type
11	Table type
12	Delete?
13	Glossary
14-18	Argument
14-16	Length
17	Data type
18	Decimal places
19-23	Result
19-21	Length
22	Data type
23	Decimal places

Table name (01-08)

Required entry

Names the table defined. See general Rule A.

Statement type (09-10)

Required entry

Identifies a table definition header statement. Enter TB.

Table type (11)

Required entry

Specifies the type of search. Enter one of the following codes:

Code	Table/Search Type
Blank or B	Binary.
D	Displacement.
S	Sequential.

Delete? (12)

Allows you to delete a table definition.

Entry	Result
Blank or N	Table definition is added to the library.
Y	Entire table is deleted from the library.
R	Definition is replaced if it is already cataloged; otherwise, it is added to the library.

Glossary (13)

Allows you to request or to suppress the table glossary.

Entry	Result
Blank or 1	Format 1 – glossary is printed.
X	Format 2 – glossary is printed.
N	No glossary is printed.

Argument (14-18)

Describes the characteristics of the table argument.

Leave blank for displacement tables.

Length (14-16)

Required entry

The maximum amount of data the argument field can contain. Enter the length within the following ranges:

Data Type	Length
Blank or C	1–255 characters.
Z	1–15 bytes.
P	1–8 bytes (accommodates up to 15 digits plus sign).
F	1–4 bytes.
E	4 bytes only.

Data type (17)

Indicates the type of data in the table argument.

Enter one of the following:

Entry	Data Type
Blank or C	Character string.
Z	Zoned decimal number.
P	Packed decimal number.
F	Fixed point binary number (negative values are stored in 2's complement notation).
E	Floating point number.

Decimal places (18)

Indicates the number of decimal places in the table argument (numeric type Z, P, F fields only).

Entry	Decimal Places	Data Type
Blank	0	C, E
0–9	0–9	Z, P, F

- For type Z argument fields, the number of decimal places must not exceed the maximum field length.
- For type P argument fields, the number of decimal places must not exceed twice the field length minus one.
- For type F argument fields, the maximum number of decimal places depends on the field length, as shown.

Length	Maximum Number
1	3
2	5
3	7
4	9

Result (19-23)

Describes the characteristics of the table result.

Note: If a table's result field characteristics are changed, the file definition must also be modified (in a definition/ maintenance run) by deleting the LA entry on the file and redefining it with the same information if using automatic table lookup.

Length (19-21)

Required entry

The maximum amount of data the table result field can contain. Enter the table result field length in the following ranges:

Data Type	Length
Blank or C	1–255 characters.
Z	1–15 bytes.
P	1–8 bytes (accommodates up to 15 digits plus sign).

Data Type	Length
F	1–4 bytes.
E	4 bytes only.

Data type (22)

Indicates the type of data in the table result field.

Enter one of the following codes:

Entry	Data Type
Blank or C	Character string.
Z	Zoned decimal number.
P	Packed decimal number.
F	Fixed point binary number (negative values are stored in 2's complement notation).
E	Floating point number.

Decimal places (23)

Indicates the number of decimal places in a numeric table result field.

Entry	Decimal Places	Data Type
Blank	0	C, E
0–9	0–9	Z, P, F

- For type Z result fields, the number of decimal places must not exceed the maximum field length.
- For type P result fields, the number of decimal places must not exceed twice the field length minus one.
- For type F result fields, the maximum number of decimal places depends on the field length, as shown.

Length	Maximum Number
1	3
2	5
3	7
4	9

TD Statement - Transaction Definition❹

Position	Entry
01-08	Transaction group name
09-10	Statement type
11-28	Transaction identifier
11-14	Code identifier1
15-18	Start location identifier1
19	Length identifier1
20-23	Code identifier2
24-27	Start location identifier2
28	Length identifier2
29	Delete?
30	Glossary
31-38	Master file name
39	Default create record?
40	Default insert segment occurrence
41-53	Transaction file characteristics
41	Record format
42-45	Record size
46-48	Records per block
49-53	Buffer size

The transaction definition header defines the following:

- Transaction group.
- Transaction identifier(s) composing the group.
- Master file definition that the transaction groups use.
- Default record creations and segment occurrence insertions.
- Characteristics of the transaction file.

Transaction group name (01-08)

Required entry

Identifies a group of transaction identifiers using one master file definition and one set of transaction file characteristics. See general Rule A. Enter an 8-character name.

Note: The same name can be used for several transaction groups, provided each uses a different master file definition.

Statement type (09-10)

Required entry

Identifies a transaction definition header statement. Enter TD.

Transaction identifier (11-28)

The transaction identifier describes the transaction code. The transaction code identifies a transaction type.

- Identifiers 1 and 2 combined allow 8 characters to identify a transaction code.
- Identifier 2 is used only if the identifier 1 code is insufficient to identify the transaction code. Identifier 2 cannot be used alone or in place of identifier 1.
- Noncontiguous fields can be used as identifiers 1 and 2.
- If only one transaction type composes a transaction group, no transaction identifier is necessary. Transaction groups that specify no identifiers are inclusive and cannot be run with other groups.
- The code must appear in every transaction data record of this transaction type for action codes that create or maintain a file.
- The code entry, whether alpha or numeric, is left-aligned; the start location and length are right-aligned.
- Transaction codes must be unique and unambiguous for all transactions in a run. If the same transaction identifier appears in two requested groups, there is no guarantee which group will be used.

Code identifier1 (11-14)
Code identifier2 (20-23)

Enter a 1–4 character code (alphanumeric) to identify the transaction data. See general Rule E.

- When designated as 0, 11, or 12 with a length of Z or X, the code is treated as a zoned representation.
- A blank code with a specified start location and length is a legitimate entry in the transaction identifier.

Start location identifier1 (15-18)
Start location identifier2 (24-27)

Identifies the location on the record of the transaction code.

Enter the start position of the transaction code. See general Rule D.

Length identifier1 (19)
Length identifier2 (28)

Specifies the number of characters in the transaction code. Transaction definitions with identical group names and codes cannot be combined in the same run (including found and not found relationships).

Entry	Description
1–4	Length of the code entry; transaction is applied by the transaction definition when the code entry is found.
Z	Code is zoned representation; transaction is applied when code is found.
A	Length is 1; transaction is applied by the transaction definition when the code is not found.
B	Length is 2; transaction is applied by the transaction definition when the code is not found.
C	Length is 3; transaction is applied by the transaction definition when the code is not found.
D	Length is 4; transaction is applied by the transaction definition when the code is not found.
X	Code is zoned representation; transaction is applied when the code is not found.

Delete? (29)

Deletes either a transaction identifier or an entire transaction group from the library.

Enter one of the following:

Delete?	Required Field Entries for a Delete	Result
Blank or N		VISION:Builder accepts the statement as a new definition or an addition to an existing one.
Y	Transaction group name, identifier(s), and master file name	Deletes the transaction identifier from the group.
Y	Transaction group name and master file name	Deletes the existing transaction group (including all identifiers).
R	Transaction group name and master file name.	Replaces an existing transaction group (including all identifiers).

Glossary (30)

Allows you to select one of three glossary listings. See Figure 2-7 through Figure 2-9 at the end of the section on the TD statement.

Entry	Resulting Glossary Within Transaction Identifier
Blank or 1	Sequentially in order of input by level number, action code.
2	Field location sequence.
3	Alphabetically by field name.
N	No glossary.

Note: A transaction glossary can be requested in a catalog maintenance run by specifying the transaction glossary name, master file name, and glossary type.

Master file name (31-38)

This entry is required if it is the first or only TD statement in the group. On subsequent TE statements, if the master file name is blank, VISION:Builder defaults to the master file name, record format, record size, records per block, and buffer size entries on the first TD statement.

Identifies the master file against which the transaction definition is applied. See general Rule A. The master file must exist in the library.

Default create record? (39)

Note: When no match is found, the entries in positions [39] and [40] permit automatic record creation and segment occurrence insertion.

Entry	Result When Requesting Master File Match
Blank or N	The transaction is rejected if no matching key is found.
Y	The record is automatically created if a match is not found.

Default insert segment occurrence (40)

Entry	Result
Blank or N	The transaction record is rejected if no matching key is found.
Y	The segment occurrence is automatically inserted if it does not exist in the record.
2 to 9	Highest level at which segment occurrence can be added.

- New segments are inserted at the level where the match to the segment key failed.
- An entry of 2 is the same as Y.

Characteristics of transaction file (record format, record size, records per block, and buffer size) (41-53)

These entries describe the characteristics common to the transaction records in the transaction file.

Record format (41)

Note: If you specify I, A, or J record format, a file definition for the transaction file must be defined, must have the same record format and record size or buffer size as the transaction definition, and must specify the key field of the VSAM/ISAM file. It does not need to be the same as the transaction definition match field.

Required entry

Entry	Format
F	Fixed length
I	Fixed ISAM
V	Variable length
J	Variable ISAM
P	Packed
K	Key sequenced VSAM
U	Undefined length
E	Entry sequenced VSAM
A	Alternate index VSAM

Record size (42-45)

See general Rule D.

Record Format	Record Size Entry
U, K, E, A	Blank.
F, I	1–9999. If left blank, you must enter the block size.
Unblocked V or P	Maximum record size (LRECL minus 4).
Blocked V, P, or J	Maximum record size. If blank, the maximum record size is computed by VISION:Builder using the buffer size specification.

Records per block (46-48)

See general Rule D.

Record Format	Entry
A, V, U, J, K, E, P, or Unblocked F	Blank.
Blocked F or I	1–999.

Buffer size (49-53)

Specifies the amount of storage required for the buffer, the block size, or the logical record.

- Enter all values right-aligned.
- Leading zeroes are not required.
- 1–32,760 or nnnnK for multiples of 1024 bytes.

Record Format	Buffer Size Entry
F, I	Enter the block size.
V, U, J (Blocked), P (Blocked)	Maximum buffer size (BLKSIZE minus 8). If variable block spanned with a record size greater than 32,760, code maximum record size.
K, E	Maximum record size (according to the VSAM define cluster specification if VSAM).
A	Alternate index control interval size.

Note: If the operating system supports system-determined block size and the file meets the criteria for this capability, this value is ignored when VISION:Builder processes the file, allowing the operating system to determine the optimum block size and allocate the buffers.

```
JAN 15, 1995  12.33.05                                                                                    PAGE 11

                                        GLOSSARY BY LEVEL NUMBER FOR                                      GLOSSARY PAGE 2
                                        TRANSACTION GROUP - ORDENTRY

                                      ************************************
                                      *    2 OF 4 IDENTIFIERS IN GROUP    *
                                      ************************************

**********  ACTIONS IN IDENTIFIER = 6          IDENTIFIER ONE = (ADD)              IDENTIFIER TWO = (ITEM)
*ADD ITEM*  DEFAULT CREATE = NO                IDENTIFIER LOCATION = 1             IDENTIFIER LOCATIO = 5
**********  DEFAULT INSERT AT LEVEL 3          IDENTIFIER LENGTH = 4               IDENTIFIER LENGTH = 4
                                               IDENTIFIED IF EQUAL                 IDENTIFIED IF EQUAL

-------------------------------------------------------------------------------------------------------------------------
MASTER  ****** TRANSACTION DATA ******
FIELD   FIELD  FIELD    FIELD   DECIMAL   ACTION   LEVEL       MINIMUM                        MAXIMUM
NAME    TYPE  LOCATION  LENGTH  PLACES     CODE    NUMBER       VALUE                          VALUE
-------------------------------------------------------------------------------------------------------------------------
CUSTNO     C     5        5                M       1          EDIT = Y9999
ORDERNO    C    14        5                M       2
ITEMORD    C    19        7                M       3
ITEMNAME   C    26       20                R       3
ITMPRICE   Z    47        6                R       3       ***0.00            ***     ***300.00            ***
ITMQTYOR   Z    60        7                R       3
```

Figure 2-7 Transaction Glossary - Type 1 or Blank

```
JAN 15, 1995  12.33.05                                                                                    PAGE 11

                                        GLOSSARY BY LOCATION FOR                                          GLOSSARY PAGE 2
                                        TRANSACTION GROUP - ORDENTRY

                                      ************************************
                                      *    2 OF 4 IDENTIFIERS IN GROUP    *
                                      ************************************

**********  ACTIONS IN IDENTIFIER = 6          IDENTIFIER ONE = (ADD)              IDENTIFIER TWO = (ITEM)
*ADD ITEM*  DEFAULT CREATE = NO                IDENTIFIER LOCATION = 1             IDENTIFIER LOCATIO = 5
**********  DEFAULT INSERT AT LEVEL 3          IDENTIFIER LENGTH = 4               IDENTIFIER LENGTH = 4
                                               IDENTIFIED IF EQUAL                 IDENTIFIED IF EQUAL

-------------------------------------------------------------------------------------------------------------------------
MASTER  ****** TRANSACTION DATA ******
FIELD   FIELD  FIELD    FIELD   DECIMAL   ACTION   LEVEL       MINIMUM                        MAXIMUM
NAME    TYPE  LOCATION  LENGTH  PLACES     CODE    NUMBER       VALUE                          VALUE
-------------------------------------------------------------------------------------------------------------------------
CUSTNO     C     5        5                M       1          EDIT = Y9999
ORDERNO    C    14        5                M       2
ITEMORD    C    19        7                M       3
ITEMNAME   C    26       20                R       3
ITMPRICE   Z    47        6                R       3       ***0.00            ***     ***300.00            ***
ITMQTYOR   Z    60        7                R       3
```

Figure 2-8 Transaction Glossary - Type 2

```
JAN 15, 1995  12.33.05                                                    PAGE 11

                                    GLOSSARY BY NAME FOR                  GLOSSARY PAGE 2
                                  TRANSACTION GROUP - ORDENTRY

                                 ************************************
                                 *    2 OF 4 IDENTIFIERS IN GROUP    *
                                 ************************************

**********  ACTIONS IN IDENTIFIER = 6          IDENTIFIER ONE = (ADD)          IDENTIFIER TWO = (ITEM)
*ADD ITEM*  DEFAULT CREATE = NO               IDENTIFIER LOCATION = 1         IDENTIFIER LOCATIO = 5
**********  DEFAULT INSERT AT LEVEL 3         IDENTIFIER LENGTH = 4           IDENTIFIER LENGTH = 4
                                              IDENTIFIED IF EQUAL             IDENTIFIED IF EQUAL

-------------------------------------------------------------------------------------------------------------------------
MASTER  ****** TRANSACTION DATA ******
FIELD   FIELD  FIELD     FIELD    DECIMAL   ACTION   LEVEL       MINIMUM                        MAXIMUM
NAME    TYPE  LOCATION  LENGTH   PLACES     CODE    NUMBER       VALUE                          VALUE
-------------------------------------------------------------------------------------------------------------------------
CUSTNO     C     5         5                M        1          EDIT = Y9999
ITEMNAME   C    26        20                R        3
ITEMORD    C    19         7                M        3
ITMPRICE   Z    47         6                R        3       ***0.00            ***     ***300.00            ***
ITMQTYOR   Z    60         7                R        3
ORDERNO    C    14         5                M        2
```

Figure 2-9 Transaction Glossary - Type 3

TE Statement - Table Definition

The TE statement defines the table argument and result entries.

Position	Entry
01-08	Table name
09-10	Statement type
11	Continuation
12	Delete?
13-42	Argument value
43-72	Result value

Table name (01-08)

Required entry

Enter the table name as on the TB statement for this table.

Statement type (09-10)

Required entry

Identifies the table entry definition statement. Enter TE.

Continuation (11)

Indicates multiple table definition entry statements for the argument and/or the result of a table.

Entry	Result
Blank	One statement for a table argument and result.
1–9	Multiple statements, consecutively numbered, for a table result or argument. The first TE statement is numbered 1.

- Continuation statements are not required for entries when the remainder of the long argument and/or result is blank.
- Multiple TE statements constituting one table entry must be in sequence by continuation number.
- A character argument or result value is determined by the number of characters (including blanks) equal to the argument or result value length specified on the table header statement. If too few continuation statements are provided to generate the correct length for the argument/result, the result field will be padded with trailing blanks.
- The character data on the TE statement is converted to the type specified on the TB statement. If continuation is specified and the conversion is to a numeric data type (Z, P, F, or E), the standard VISION:Builder scanning conventions apply. Leading sign, embedded commas, and decimal points are interpreted correctly. The scan terminates with the first trailing nonnumeric character.

Delete? (12)

Allows you to delete an existing table entry.

Enter Y to delete an existing table entry. Enter N or leave blank and VISION:Builder will add the entry to the table in the library.

Argument value (13-42)

Required entry

Specifies the table argument values.

Enter the character representation of the data value you want in the table (with leading signs and decimal points when needed). The value can be up to 255 bytes long and will be converted internally to the argument described on the TB statement. Blank can be a valid argument. Only the integer portion of the argument value is used for displacement tables.

Note: If argument and/or result fields are defined on the TB statement as having decimal places and no explicit decimal point is entered in the actual value, VISION:Builder assumes the decimal point to be at the right of the field.

Result value (43-72)

Required entry

Describes the returned value for the table corresponding to the table argument.

Enter the character representation of the data value you want in the table (with leading signs and decimal points when needed). The value can be up to 255 bytes long and will be converted internally to the result described on the TB statement. Blank can be a valid result.

Note: If argument and/or result fields are defined on the TB statement as having decimal places and no explicit decimal point is entered in the actual value, VISION:Builder assumes the decimal point to be at the right of the field.

TF Statement - Temporary Field Statement

Position	Entry
01-08	Request name
09-10	Statement type
11-18	Field name
19-21	Field length
22	Field type
23	Decimal places
24-28	Output edit
24	DBCS data state
24	Floating/edit suppress
25	Filling
26	Trailing
27-28	Length
29-44	Initial value
45-58	Line1 column heading text
59-72	Line2 column heading text

The temporary fields statement (TF statement) creates temporary fields, which allows you to perform computations, output computed values, communicate between requests and records, retain constant values for use in request processing, and store intermediate results.

Temporary fields are created automatically when a T qualifier is entered on the processing and record selection statement and the field name has not been previously defined as a temporary field. The name of the temporary field must conform to general Rule A.

Temporary fields can be explicitly or implicitly defined.

Explicitly defined

Explicitly defined temporary fields use the temporary field definition statement. The TF statement enables you to specify the length and type of the field, output editing, initial values, and up to 2 lines of column heading text.

Implicitly defined

When a temporary field is created without having been explicitly defined, VISION:Builder assigns it the attributes of the base field. The base field is the leftmost defined field in the PR statement where the temporary field first appears. C and D constants can be base fields because they are defined fields. If there are no defined fields in the statement, the temporary field takes the following default attributes:

Field length	8 bytes
Field type	P (packed decimal)
Field rounding	Blank
Decimal places	8
Output edit length	19 characters
Leading edit characters	Blank
Filling edit characters	Blank
Trailing edit characters	Blank

Request name (01-08)

Required entry

Identifies this statement and the others accompanying it (if any) as one request. See general Rule A. Enter the same name for all statements in the request.

Statement type (09-10)

Required entry

Identifies a temporary field definition statement. Enter TF.

Field name (11-18)

Required entry

Identifies the symbolic name of the field defined. See general Rule A. Enter a unique 8-character name for each temporary field; however, the name can be the same as that of a field in any file definition associated with the run.

Field length (19-21)

Required entry

Specifies the length of a field. Enter the field length within the following ranges:

<table>
<tr><th>Field Type</th><th>Range</th></tr>
<tr><td>Blank or C</td><td>1–255 bytes (characters).</td></tr>
<tr><td>V</td><td>1–999 or 1H–99H (H implies 00), excluding a 4-byte header.</td></tr>
<tr><td>Z</td><td>1–15 bytes.</td></tr>
<tr><td>P</td><td>1–15 bytes (29 digits plus sign).</td></tr>
<tr><td>L

S</td><td>Min. (byte) Max. (bytes)

$\left[\frac{N_n + 7}{2}\right]$ 8

$\left[\frac{N_n + 5}{2}\right]$ 8

N_n = number of decimal of seconds digits ($N_n < 9$).
[ ] = use only the integer portion of number.</td></tr>
<tr><td>E</td><td>4 bytes only.</td></tr>
<tr><td>F</td><td>1–4 bytes with ranges as follows:
<table><tr><th>Field Length</th><th>Range</th></tr><tr><td>1 byte</td><td>-128 to 127</td></tr><tr><td>2 bytes</td><td>-32,768 to 32,767</td></tr><tr><td>3 bytes</td><td>-8,388,608 to 8,388,607</td></tr><tr><td>4 bytes</td><td>-2,147,483,648 to 2,147,483,647</td></tr></table></td></tr>
<tr><td>D</td><td>4 bytes only (interpreted as a Lilian date when displayed).</td></tr>
</table>

If the field length entry is left blank, the default lengths for each field type include the following:

Field Type	Length
C	16
Z	15
P	8
F	4
E	4
L	8
S	8
D	4

Field type (22)

Code indicating the type of data in the field.

Code	Field Type
C	Character string.
Z	Zoned decimal.
P	Packed decimal.
F	Fixed point binary (negative values are stored in 2s complement notation).
E	Floating point.
V	Variable length field.
L	Time (hours, minutes, seconds).
S	Time (minutes, seconds).
D	Date (Lilian).

Decimal places (23)

Number of implied decimal places in a numeric field.

Entry	Result
Blank	Zero decimal places assumed. Blank is the only valid entry for type V or C fields.
0–9	Number of implied decimal places.

Note: For time fields (field types L and S), this entry represents decimal positions of seconds. See general Rule D.

Output edit (24-28)

Use the output edit entries (floating, filling, trailing) only with numeric fields (types P, Z, and F). These entries edit the data before reporting it. Commas print in these fields where they are preceded by a significant digit. To suppress printing of commas, use the edit suppress character Z or the override picture edit on the Rn statement [53-67].

Make no entry in the output edit columns for floating point numbers. They always print in scientific notation. The output edit length must exceed six to include two signs, decimal point, E, and two exponent digits. Standard notation for floating point numbers, where Xs represent the fraction and Ys the exponent, is:

```
± .XXXXXXXE ± YY
```

DBCS data state (24)

This entry is valid only for character field type C in position [22] to indicate the DBCS data state.

Code	Data State
Blank	EBCDIC.
I	DBCS data with shift characters embedded in the data. This data is submitted to the printer without modification.
E	DBCS data that does not contain any embedded shift characters. This data is submitted to the printer with shift characters embedded in the data.

Floating/edit suppress (24)

An entry other than Z in this column “floats” the value of the entry and prints to the immediate left of the first significant digit in the printed report.

If this column is blank, a leading blank prints if positive, a minus if negative. If this column contains a Z, all commas, leading zeroes to the left of the decimal point, and the decimal point (if specified) are suppressed.

This entry is valid only for numeric field types P, Z, and F in position [22].

Code	Result
Blank	When no output edit codes are specified, a leading blank prints if the value of the field is positive; a leading minus sign prints if the value of the field is negative; commas print and the decimal point prints if decimal places are specified. A zero value in a field where a decimal place is specified prints as a decimal point followed by as many zeroes as there are decimal places.
$	A floating dollar sign prints before the first value when a control break occurs and when summaries are taken.
+	A leading plus sign (+) prints if the value of the field is positive, minus sign (-) if negative.
-	A leading minus sign (-) prints if the value of the field is negative (default specification).
Z	The printing of all commas, leading zeroes to the left of the decimal point, and the decimal point (if specified) are suppressed.
Any other character	A floating leading character prints on control breaks and on summaries.

Note: Negative signs print but no space is allocated for them.

Filling (25)

The value of this entry prints in every position from the leftmost of the field until the first nonzero digit is encountered. This entry is valid only for P, Z, and F fields.

Code	Result
Any character	Replaces leading zeroes.

Trailing (26)

The value of this entry prints following positive and/or negative values. This entry is valid only for P, Z, and F fields.

Code	Result
+	Prints a trailing plus sign (+) if the value of the field is positive, minus sign (-) if negative.
-	Prints a trailing minus sign (-) if the value of the field is negative.
)	Encloses negative field values in parentheses. If no filling character is specified, the left parenthesis prints before the first significant digit or decimal point, whichever comes first. If a floating character and this character are specified, both can print. The floating character prints inside the parentheses ($43.50). Only a single floating sign is permissible with the trailing ")".
C	Prints a trailing "CR" for a negative value. Blanks follow a positive value.
D	Prints a trailing "DB" for a negative value. Blanks follow a positive value.
Any other character	Prints a trailing character for negative values only.

Length (27-28)

The number of print positions for reporting a field.

Code	Result
Blank	Normal - length is computed by VISION:Builder. Type V fields - length is computed by VISION:Builder to be the shorter of either the defined field length or report page width minus spaces before columns.
Any 2-digit number	Normal - length used by VISION:Builder is the output edit length of field, including any floating, filling, and/or trailing characters. Type V fields - width of the column. Text automatically folds until it is exhausted.

Initial value (29-44)

Initialized value of the temporary field.

Code	Field Type	Initialized To
Blank	Character	Blanks
	Numeric	Zeroes
	Variable	Null

Note: Time value can be entered for field types L or S.

Line1 column heading text (45-58)
Line2 column heading text (59-72)

Entry	Resulting Column Heading and Maximum Allowed
Blank	Field name as defined in [11-18].
1–14 characters each line (up to 2 lines)	Single or double line column heading.
Delimiter	Signifies the end of a line of text if it is the rightmost nonblank character.

Note: To force a blank column heading, enter a blank and the delimiter in the first line of column heading text.

TL Statement - Transaction Definition❹

The TL statement defines actions to be applied to a master file with the transaction data. It can add new data or modify existing data.

Each field name in a TL statement refers to a master file field. The action code specifies the operation performed on the master file, such as create/delete master file record, insert/delete segment occurrence, add/subtract the contents of a field, and replace fields in the master file.

Position	Entry
01-08	Transaction group name
09-10	Statement type
11-14	Transaction identifier1
15-18	Transaction identifier2
19	Delete?
20-27	Transaction data fields
20-23	Location
24-26	Length
27	Type
28-35	Master file field name
36	Action code
38	Transaction data decimal places
39-68	Automatic validation
39-53	Minimum
54-68	Maximum
69	Validation type

Transaction group name (01-08)

Required entry

Refers to the transaction group named on the TD statement. See general Rule A.

Statement type (09-10)

Required entry

Identifies a transaction definition statement. Enter TL.

Transaction identifier1 (11-14)
Transaction identifier2 (15-18)

Enter the transaction identifier(s) identical to those on the TD statement. See general Rule E.

Note: Up to 255 TL statements are allowed for each transaction identifier.

Delete? (19)

Deletes line entries in the transaction definition.

Entry	Result
Blank or N	The line entry is added to the transaction definition.
Y	The line entry is deleted.

Note: You must delete an existing line entry before you enter a revised one.

Transaction data field (location, length, type) (20-27)

These entries describe the transaction data.

Location (20-23)

Required entry

Physically locates a transaction data field within the transaction record.

Entry	Result
1–9999	Identifies the location of the first character (byte) of the field on the transaction record.

Length (24-26)

Required entry

Specifies the length (in bytes) of the field defined on the transaction record. The number refers to the field length on the transaction record. It has no relationship to the length of the corresponding master file field and can be different from that in the file definition. The permissible field lengths include the following:

Field Type	Length
Blank or C	1–255 bytes (characters).
Z	1–15 bytes.
P	1–8 bytes (number of bytes, not characters in the field - up to 15 digits plus sign).
F	1–4 bytes.
E	4 bytes only.
L	3 to 8 bytes.
S	2 to 8 bytes.

Field Type	Length
U	5 to 15 bytes.
D	4 bytes only.

Type (27)

Identifies the data type of the field being defined.

Entry	Field Type
Blank or C	Character.
Z	Zoned decimal number.
P	Packed decimal number.
F	Fixed point binary number (negative values are stored in 2's complement notation).
E	Floating point number.
L	Time (see general Rule L).
S	Time (see general Rule L).
U	Time (see general Rule L).
D	Date (Lilian).

- This entry has no relationship to the master file field and can be different from the one specified in the file definition.
- VISION:Builder automatically converts the transaction field type to that of the master file field before performing the required action. When the updated master file entry is a type V field, the field type must be C.
- Character fields (C or blank) containing numeric data are converted to other field types.

Master file field name (28-35)

Required entry

Identifies the master file field being updated. Enter the field name exactly as it appears on the master file definition.

Action code (36)

Required entry

Identifies the action to be performed on the master file field specified in [28-35].

Entry	Result
M	Match the transaction data field to the master file field specified.
C	Create a new master file record. VISION:Builder initializes the record with zeroes or blanks (depending upon field type specified in the master file definition) in the fields not specified as the record key. Additional actions can be defined to enter data into these fields.
D	Delete an entire record from the master file.
I	Insert an entire segment occurrence in the master file record.
E	Delete an entire segment occurrence (this automatically includes all subordinate segments) from the master file record.
A	Add the contents of the transaction field to the contents of the master file field.
S	Subtract the contents of the transaction field from the contents of the master file field.
P	Unconditionally replace the specified master file field with the transaction field. If the master file field is type V, trailing blanks are eliminated after the transaction data has been applied.
R	Replace the specified master file field with the transaction field only if the transaction field is not empty. If the master file field is type V, trailing blanks are eliminated after the transaction data has been applied.
B	Replace the contents of a character master file field with blanks if the transaction record field is not empty. If the transaction field is empty, the master file record is not affected. Empty type C fields are blanks; empty type Z fields are either zero or blank. If this action is performed on a numeric field, the field is cleared to zero. If the master file field is type V, the length of the field becomes zero and all data is removed from that position of the record if the transaction field data is nonblank.
X	Transaction data is concatenated with the end of the type V master file field. Leading blanks in the transaction data are not eliminated but trailing blanks are.

- If default create/insert [39-40] is specified on the TD statement and the field in the transaction data to be matched does not already exist, the record/segment will be added to the master file record with that key.
- If default create/insert [39-40] is specified on the TD statement and you are matching on nonkey fields, you can create records or segments with duplicate keys.
- For relational tables, matching on nonkey fields is not allowed using a transaction definition. You can use an instream definition to define the field as a key or compare in a type 2 request.Ⓓ

Transaction data decimal places (38)

Specifies the number of decimal places in numeric (types Z, P, F) fields in the transaction.

Entry	Number of decimal places
Blank	0 (assumed by VISION:Builder)
0–9	0–9

- When the transaction field is applied to the master file field, the decimal places are truncated or extended to agree with the master file format.
- When adding to or subtracting from the master file field, the computation is performed after the decimal places have been aligned.

Automatic validation (39-68)

Specifies the validation pattern used in validating a character string (type C) field. When you enter a validation pattern and insert a P in the validation type entry [69], VISION:Builder automatically validates the field to your specifications. The transaction data record is rejected when the field does not conform to requirements established by the pattern.

Minimum/maximum checking can be specified with another TL statement referencing the same data field; however, the action codes are performed for each of the TL statements, in that sequence.

Minimum (39-53)
Maximum (54-68)

Constants that define the range of values for a field in the transaction. See general Rule E.

- If the specifications are blank, no checking is done.
- The minimum value or maximum value can be used by itself.
- Using both entries ensures that the value in this field of the transaction data record is in the specified range before the data is applied to the master file field.
- The transaction data record is rejected when the value of this field is larger than the maximum or less than the minimum.
- Each entry is a character string; negative values must be identified with a leading minus sign.
- Decimal points must be explicitly entered.
- Entries must be left-aligned and are limited to 15 bytes. For larger fields, enter the high order (leading) 15 bytes.
- Minimum/maximum transaction editing cannot be specified for type V fields.

Note: Leave the minimum/maximum value entries blank for automatic date validation.

Validation type (69)

Specifies whether validation is performed for a pattern or a date.

Entry	Result
Blank	No validation.
P	Input validation pattern. The validation symbols used to construct validation patterns include the following: Symbol — Character Set Represented 9 — Numeric (0–9). a — Alpha (A–Z, a–z) or blank. A — Alpha (A–Z) or blank. B — Blank. C — Any character (no validation). D — System delimiter. I — Numeric (0–9) or blank. x — Alpha (A–Z, a–z), numeric (0–9), or blank. X — Alpha (A–Z), numeric (0–9), or blank. y — Alpha (A–Z, a–z) or numeric (0–9). Y — Alpha (A–Z) or numeric (0–9). z — Alpha (A–Z, a–z). Z — Alpha (A–Z). Literals — Any character(s) except the minus sign (-) can be scanned for. User defined — Provided by you at system installation time (see the *VISION:Builder Reference Guide*).
D	Date validation.

- A minus sign (-) before any of these characters means VISION:Builder scans for other than the specified pattern. One minus sign operates on only one validation symbol or literal (that is, -99 is a nonnumeric character followed by a numeric character, not two nonnumeric characters).
- System delimiters must surround literals in a scan validation symbol.
- The transaction record is rejected when this field does not conform to the rules governing date validation (see the *VISION:Builder Reference Guide*).
- For automatic validation, the minimum/maximum entries [39-68] must be blank.

Tn Statement - Title/Preface/Formatted Reporting

The Tn statements specify the title to be printed on all numbered pages after the preface page.

If the Tn statement is used in a report that produces a graph, see Chapter 5, *Graphic Fixed Format Statement Listing* for the Tn statement specifications for graphics output.

Position	Entry
01-08	Request name
09-10	Statement type
11-13	Sequence number
14-72	Text

Request name (01-08)

Required entry

Identifies the request to which this statement belongs. See general Rule A. Enter the same request name for all statements in the same request.

Statement type (09-10)

Required entry

Identifies a title statement. Enter a T followed by a number (n) from 1–9. The number must be the same on all En, Kn, Rn, Pn, and Tn statements in a graph report statement group.

Sequence number (11-13)

Specifies the sequence of the report title lines. See general Rule I.

If blank lines are needed in the title, specify a sequence number and the delimiter. Duplicate sequence numbers are accepted in order of input.

Entry	Result
Blank	Order of input is assumed.
Up to 3 alpha, numeric, or alphanumeric characters	Order of report title lines. Sequencing is in ascending order.

Text (14-72)

Specifies the report title. Enter the title line-by-line.

- VISION:Builder scans a line of text until the delimiter is encountered. (The scan continues over several lines, if necessary.) The line of text ends at the first occurrence of the delimiter.
- Title lines are normally centered at the top of the page between the date and page number. However, this default can be overridden by an entry on the En statement.
- Only 59 characters of text are defined per statement. It can be necessary to use multiple statements to define a title with text that is longer than 59 characters.

UC Statement - Library Maintenance

The utility control statement (UC statement) provides capabilities that allow you to establish and maintain your library. You can perform the following operations:

Position	Entry
01-08	Run name
09-10	Statement type
11-14	Operation code
15	Generic item name
16-17	Item type code
18-25	Item name
26-33	Master file name ❹
34-41	Library DTF/ddname
42	Replace option?
43-45	Replace rule
46-49	Directory blocking factor
50	Purge expired items?

- Initialize libraries.
- Create a backup copy of a library.
- Reload a library from a backup copy.
- Condense (reorganize) a library.
- Copy items from one library to another.
- Merge all items in one library to another library.

The UC statement is used in conjunction with the MARKUTIL program.

No other statement types are valid for a MARKUTIL execution, including the RC statement.

Multiple UC statements can be included in a MARKUTIL job step.

Run name (01-08)

Optional entry. Will be ignored if entered.

Statement type (09-10)

Required entry

Identifies a utility control statement. Enter UC.

Operation code (11-14)

Identifies the type of library operations to be performed.

Entry	Result
INIT	Initializes an M4LIB.
DUMP	Creates a backup copy of library.
REST	Restores library from backup.
COND	Reorganizes library.
COPY	Copies stored definitions from M4LIBn (n=1–9) to M4LIB.
MERG	Copies entire contents of M4LIBn (n=1–9) to M4LIB.

Note: Expired items are not restored if position [50] (purge expired items) contains a Y.

Generic item name (15)

Identifies whether the item name is a full name or generic. This entry is used with the COPY operation.

Entry	Result
Blank	Item name is the full item name.
Y	Item name is a generic name (partial name).

Item type code (16-17)

Identifies the item type to be processed. This entry is used with the COPY operation.

Entry	Result
Blank	All item types.
AD	Array definition.
DB	Run data group.
FD	File definition.
RQ	Request.
RG	Request group.
TB	Table definition.
TD	Transaction group definition.❹

Item name (18-25)

Enter the full or generic item name left-aligned. This entry is used with the COPY operation. This entry is required if item type code [16-17] is blank.

Master file name❹ (26-33)

Identifies the master file name if a transaction group definition is the item type. This entry is used with the COPY operation.

Entry	Result
Blank	Item type code is other than TD.
Full or generic master file name	Master file name associated with the identified transaction group.

Library DTF/ddname (34-41)

Identifies the library to be processed.

Entry	Result
Blank	Default library for the particular operation code includes the following: INIT Initialize M4LIB. DUMP Dump M4LIB. REST Restore M4LIB. COND Condense M4LIB. COPY Copy from M4LIB1. MERG Merge entire contents from M4LIB1.
M4LIBn (n=1–9)	Identifies a particular M4LIBn to be processed.

Note: The library into which items are copied or merged is always M4LIB.

Replace option? (42)

Can be used with either the COPY or MERG operation.

Entry	Result
N or blank	Suppress the copy or merge operation if the item exists on M4LIB.
Y	Replace items on M4LIB with like-named items from the sending library according to the replace rule [43-45].

Replace rule (43-45)

Identifies type of replacement to take place when the replace option is chosen.

Entry	Result
Blank	Replaces item on M4LIB with like-named item found in another library.
OLD	Replaces item on M4LIB with a like-named item that has been less recently updated from another library.
NEW	Replaces item on M4LIB with a like-named item that has been more recently updated from another library.

Note: If update information is not available, the operation fails and the run continues with the next UC statement (if any).

Directory blocking factor (46-49)

Valid only with INIT, COND, and REST operation.

Enter the number of entries in a library directory block, right-aligned (BDAM only).

Entry	Result
Blank	Specification from MARKLIBP is used.
1–9999	Number of entries in a library directory block.

- If MARKLIBP directory blocking factor is zero, VISION:Builder calculates a directory blocking factor based on the number of tracks allocated to the library and the track size of the device upon which the library is being defined.
- If the blocking factor specified is too large for the DASD on which the library resides, the directory blocking factor will be reduced to the largest factor allowed for the DASD.
- Limits for each support device are provided in MARKLIBP.

Purge expired items? (50)

Valid only with REST operation.

Entry	Result
N or blank	Restores items with expired dates, along with all other items, when a RESTORE operation is performed.
Y	Purges all items that have expired dates when a RESTORE operation is performed.

WH Statement - Run Control Group

The where (WH statement) provides the option of extending the selection criteria of logical relationship (LR) statements for segments in a relational table.

If your COMLIB system is at the proper maintenance level to support the multi-column keyed segment capability, WH statements must be provided by the user for the necessary key comparisons for each multi-column keyed segment.

Position	Entry
01-08	Run name
09-10	Statement type
11-18	Segment name
23	Qualifier
24-71	Statement

Run name (01-08)

Required entry

Identifies this statement as belonging to a particular run group. Enter the name as entered on the RC statement.

Statement type (09-10)

Required entry

Identifies a where statement. Enter WH.

Segment name (11-18)

Required entry

Enter the name of the segment for which you will supply selection criteria. The segment must be defined in a file supplied on the RC or RF statement.

Qualifier (23)

Required entry

Identifies the input file.

Code	Location of File
0, O, or blank	Master file.
1–9	Coordinated files 1–9.

Statement (24-71)

Required entry

Enter supplemental WHERE clause conditions for this segment. The statement can also include UNION, UNION ALL, GROUP BY, ORDER BY, and HAVING clauses. This text must conform to DB2/SQL syntax rules. The text can be continued over multiple statements, if necessary. A change in segment name and qualifier determines the end of the text for a set of WH statements.

If UNION, UNION ALL, GROUP BY, ORDER BY, and HAVING clauses are specified, they should begin on a new statement (line) so that VISION:Builder can recognize the presence of these clauses and avoid generating duplicate clauses.

- If GROUP BY is the first element on any WH statement, VISION:Builder suppresses the GROUP BY clause generated whenever a column function is involved.
- If ORDER BY is the first element on any WH statement, VISION:Builder suppresses the generation of its ORDER BY clause.

If the segment named in a WH statement with a UNION clause has order specified in the file definition, the ORDER BY clause generated by VISION:Builder must be suppressed. The user must either change the file definition to eliminate ordering or supply the correct ORDER BY clause using column numbers in place of column names.

If the segment named in a WH statement is a multi-column keyed segment, VISION:Builder does not generate an ORDER BY clause. The user must supply the ORDER BY clause to order rows.

If UNION and UNION ALL clauses are specified, you must ensure that the SELECT statement for the table being combined with the SELECT statement generated by VISION:Builder has the identical number of columns in the SELECT and that these columns are of compatible data types. Because the SELECT statement generated by VISION:Builder is optimized to only include those columns referenced in the application, a simple change to a request in the application may cause the SELECT statement following the UNION to become invalid.

Host variables can be used in a WH statement to refer to a VISION:Builder field. The general format is:

```
:QUALIFIER.FIELDNAME
```

where fieldname is the name of a VISION:Builder field. It cannot be a type V field or an automatic table lookup result field.

Refer to the *VISION:Builder Reference Guide* for more information on how to construct the supplemental WHERE statements for a multi-column keyed segment defined in a coordinated file.

Wn Statement - Output Specification

This statement provides the information needed for an extended subfile that is a relational table.

Position	Entry
01-08	Run name
09-10	Statement type
27	Table control
30-37	Authorization ID
38-55	Table name
56-63	Database name or DBSPACE owner
64-71	TABLESPACE or DBSPACE

Run name (01-08)

Required entry

Identifies this statement as belonging to a particular run group. Enter the same name as entered on the RC statement.

Statement type (09-10)

Required entry

Enter a W followed by a number (n) from 1-9 to relate this statement to a specific output group.

Table control (27)

Required entry

Entry	Result
C	Create - creates a table and inserts all entries into the created table. If the table already exists, an error message is issued.
D	Delete/Replace - deletes all existing rows from a table and inserts new rows. If the table does not exist, an error message is issued.
I	Insert - inserts all entries into the existing table. If the table does not exist, an error message is issued.
R	Drop/Replace - drops the existing table, creates a new table, and inserts new rows. If the table does not exist, it is created and the rows are inserted. Caution: The action R causes a DROP that will lose all synonyms, authorizations, and so on, associated with the table.

Note: These actions can only take place if the application user has proper authorization provided to DB2 by the system administrator.

Authorization ID (30-37)

The authorization ID of the owner of the table. This specification takes precedence over any authorization ID specification on the RF statement.

- If an authorization ID is provided on either the RT or RF statement, VISION:Builder supplies DB2 with a prefixed table name in the form of:

  ```
  AUTHORIZATIONID.TABLENAME.
  ```

- If an authorization ID is not provided, VISION:Builder supplies DB2 with an unprefixed table name. In this case, DB2 uses the current SQL authorization ID. This comes from the SQLID parameter provided on the RP statement in the application.
- If the RP statement is not provided, DB2 uses the signon user ID as the table prefix. Under OS/390, this is entered on the JOB statement. Under VSE and CMS, this is entered on the RP statement.

Table name (38-55)

Required entry

Enter the name of the table as it is to be known to DB2.

Database name or DBSPACE owner (56-63)

For DB2, enter the database name. If left blank, the system default takes effect.

For SQL/DS, enter the name of the owner of the DBSPACE. If this entry is left blank, the system default takes effect. If an entry is made here, an entry must also be made for DBSPACE [64-71].

TABLESPACE or DBSPACE (64-71)

For DB2, this entry identifies the TABLESPACE name. If left blank, the system default takes effect.

For SQL/DS, this entry identifies the DBSPACE name. If left blank, the system default takes effect.

For TSO ATTACH and CALL ATTACH, if this entry is left blank and R is specified for table control [27], VISION:Builder calls DB2 to perform a COMMIT after the DROP, prior to the CREATE.

Chapter 3

Flags

A flag is an internal indicator defined by VISION:Builder to indicate the existence of certain conditions during the application run. Special words function as flags. The names of the flags are not reserved; they can be used as field names without restriction. Flags are used in processing and reporting statements. The F qualifier identifies a flag to VISION:Builder.

Flag	Application	Format
ASTATUS	Indicates the status of an array operation.	4 bytes character
CHKP[1]	Directs VISION:Builder to take a checkpoint before reading the next master file root segment.	1 byte character
CKPTID[1]	Contains the checkpoint ID of the last checkpoint taken.	8 bytes character
COLUMN	Indicates the number of the current column of the array.	4 bytes fixed
COMMAND[G]	Contains the command identifier.	8 bytes character
CONDCODE	For OS/390 and CMS, modifies VISION:Builder condition codes; for VSE, terminates job at end of job step. (User modifiable)	2 bytes binary
CSTATUS	Indicates that a CALL was suppressed by VISION:Builder and gives the reason.	4 bytes character
DATE	Records the operating system date in the format: MMM DD, YYYY.	12 bytes character
DELETE[4]	Directs VISION:Builder to delete the current master file record, to delete all segment occurrences having empty key field values, or to reject the current transaction.	1 byte fixed

Flag	Application	Format
ECORD	Indicates the match condition between the master record key field and the key fields of the current coordinated records.	9 bytes character
EOF	Lets you detect or force end of file on any sequentially input files. (User modifiable)	11 bytes character
FDNAME G	Contains the file name of the mapped file.	8 bytes character
FILE G	Contains the VISION:Builder logical file name.	8 bytes character
FILEID G	Contains the file identification of the mapped file.	8 bytes character
ISDATE	Records the operating system date in the format:	
	■ YYYYMMDD (normal usage)	8 bytes character
	■ YYYY-MM-DD (formatted reports)	10 bytes character
JULANX	Records the operating system date in the format:	
	■ YYYYDDD (normal usage)	7 bytes character
	■ YYYY.DDD (formatted reports)	8 bytes character
JULIAN	Records the operating system date in the format:	
	■ YYDDD (normal usage)	5 bytes character
	■ YY.DDD (formatted reports)	6 bytes character
LILIAN	Date type field containing the Lilian date as the number of days since the beginning of the Gregorian calendar (October 14, 1582). The valid range of Lilian dates is 1 – 3,074,324 (October 15, 1582 to December 31, 9999). For example, a Lilian date with a value of 152384 converts to the standard date of December 31, 1999.	4 bytes integer (date type)

Flag	Application	Format
LNUMBER	Indicates the number of characters in the left part of the field just scanned. (User modifiable)	4 bytes fixed
LSTART	Indicates the starting location of the left part of the field just scanned. (User modifiable)	4 bytes fixed
LSTATUS	Indicates the status of segment operations.	4 bytes character
M4AUDIT❹	Provides the number of deleted master file records output to M4AUDIT during application execution.	4 bytes fixed
M4CORD1	Provides the number of records read from M4CORD1 during application execution.	4 bytes fixed
M4CORD2	Provides the number of records read from M4CORD2 during application execution.	4 bytes fixed
M4CORD3	Provides the number of records read from M4CORD3 during application execution.	4 bytes fixed
M4CORD4	Provides the number of records read from M4CORD4 during application execution.	4 bytes fixed
M4CORD5	Provides the number of records read from M4CORD5 during application execution.	4 bytes fixed
M4CORD6	Provides the number of records read from M4CORD6 during application execution.	4 bytes fixed
M4CORD7	Provides the number of records read from M4CORD7 during application execution.	4 bytes fixed
M4CORD8	Provides the number of records read from M4CORD8 during application execution.	4 bytes fixed
M4CORD9	Provides the number of records read from M4CORD9 during application execution.	4 bytes fixed
M4NEW❹	Provides the number of records output to M4NEW during application execution.	4 bytes fixed
M4OLD	Provides the number of records read from M4OLD during application execution.	4 bytes fixed
M4REJECT❹	Provides the number of records output to M4REJCT during application execution.	4 bytes fixed
M4SUBF0	Provides the number of records output to M4SUBF0 during application execution.	4 bytes fixed

Flag	Application	Format
M4SUBF1	Provides the number of records output to M4SUBF1 during application execution.	4 bytes fixed
M4SUBF2	Provides the number of records output to M4SUBF2 during application execution.	4 bytes fixed
M4SUBF3	Provides the number of records output to M4SUBF3 during application execution.	4 bytes fixed
M4SUBF4	Provides the number of records output to M4SUBF4 during application execution.	4 bytes fixed
M4SUBF5	Provides the number of records output to M4SUBF5 during application execution.	4 bytes fixed
M4SUBF6	Provides the number of records output to M4SUBF6 during application execution.	4 bytes fixed
M4SUBF7	Provides the number of records output to M4SUBF7 during application execution.	4 bytes fixed
M4SUBF8	Provides the number of records output to M4SUBF8 during application execution.	4 bytes fixed
M4SUBF9	Provides the number of records output to M4SUBF9 during application execution.	4 bytes fixed
M4TRAN ❹	Provides the number of records read from M4TRAN during application execution.	4 bytes fixed
MISSPASS	Indicates the status of the follow-up pass in sequential coordination for master file records.	1 byte character
MNUMBER	Indicates the number of characters in the middle part of a field just scanned. (User modifiable)	4 bytes fixed
MODE Ⓖ	Contains information about the application mode of operation.	2 bytes character
MSTART	Indicates the starting location of the middle part of a field just scanned. (User modifiable)	4 bytes fixed
MSTATUS Ⓖ	Used by the mapping request to instruct VISION:Builder on a specific action to take subsequent to completion of the mapping request. (User modifiable)	6 bytes character
OWN	Provides a means of communication between user routines and own-code exits. (User modifiable)	16 bytes character

Flag	Application	Format
PAGE	Indicates placement of page numbers in formatted reporting.	6 bytes character
PASSWORD[G]	Contains the password from the RF statement.	8 bytes character
RESTART[1]	Indicates restart ID or blank if not a restart.	8 bytes character
RETURNCD	Determines the results of the CALL as set by the generalized system interface CALL routine.	4 bytes fixed
RNUMBER	Indicates the number of characters in the right part of a field just scanned. (User modifiable)	4 bytes fixed
ROW	Indicates the row number in the array.	4 bytes fixed
RSTART	Indicates the starting location of the right part of the field just scanned. (User modifiable)	4 bytes fixed
RSTATUS	Indicates the results of a read operation.	4 bytes character
SEGNAME[G]	Contains the segment name as defined on the LS statement.	8 bytes character
SQL[D]	Indicates the status of inserting a row into an SQL table.	4 bytes character
SSCOUNT	Counts the number of matches found during a REPLACE operation.	2 bytes fixed
STRAN[4]	Interrogates status of transactions at each segment and indicates which have been applied to the master file.	9 bytes character
TIME	Records the time of day a job was started in the format: HH.MM.SS (hours, minutes, seconds).	8 bytes character
TODAY	Records the operating system date in the format:	
	■ MMDDYY (normal usage)	6 bytes character
	■ MM/DD/YY (formatted reports)	8 bytes character

Flag	Application	Format
TODAYX	Records the operating system date in the format:	
	■ MMDDYYYY (normal usage)	8 bytes character
	■ MM/DD/YYYY (formatted reports)	10 bytes character
TRAN[4]	Indicates the status of a master file record and/or the rejection of a transaction.	1 byte fixed
XTRAN[4]	Identifies the reason for rejection of a particular transaction.	1 byte fixed

ASTATUS Flag

The ASTATUS flag indicates the status code following the execution of an array operation. It is a character field with a length of 4 bytes. The status information can be examined to determine if the operation was successful, failed, or was suppressed.

- The ASTATUS flag can be useful in debugging applications or in determining sources of erroneous input to applications.
- As part of the array operation, there is an implied compare to blanks in the ASTATUS flag when there is an NS or GS operation immediately following an array operation. The NS or GS branch will be taken on any status other than blanks. At the "branch-to" location, the status information can be examined to determine why the operation failed or was suppressed.

The ASTATUS codes and explanations are shown in the following table.

	Operations Involved				
Value	Locate Row and Column	Locate Row	Locate Column	Release	Comments
b/b/b/b/	Yes	Yes	Yes	Yes	The operation executed successfully.

	Operations Involved				
Value	Locate Row and Column	Locate Row	Locate Column	Release	Comments
BMIS BINV CMIS CINV	Yes Yes Yes Yes	Yes Yes No No	No No Yes Yes	No No No No	The operation is unsuccessful. The value of the indicated row or column is missing or invalid as indicated or is larger than a 4-byte fixed point field. If the first character is B, the error applies to the row. If it is C, the respective error applies to the column.
ROWH ROWL COLH COLL	Yes Yes Yes Yes	Yes Yes No No	No No Yes Yes	No No No No	The operation is unsuccessful. The value supplied for row or column is not within the dimensions of the array. The value is greater than the highest row number. The row value is less than 1. The value is greater than the highest column number. The column value is less than 1.
AUTO	Yes	Yes	Yes	Yes	The operation is suppressed. An automatic loop is active on the array.

	Operations Involved				
Value	Locate Row and Column	Locate Row	Locate Column	Release	Comments
INHR	Yes	Yes	Yes	Yes	The operation is suppressed. A subroutine was called; it attempted to perform an array operation on an array that was already positioned to a specific data cell due to field references in the calling request, or a subset of the array was located by a calling request by means of a successful array operation.

CHKP Flag

The CHKP flag triggers a checkpoint operation by placing a non-blank value into the flag from any request except preselection (type P). It is a character field with a length of 1 byte. The checkpoint does not occur until just before VISION:Builder is ready to read the next master file root segment. The flag is reset to a blank after every checkpoint.

The user can place an "A" in the CHKP flag to force an ABEND to occur instead of a checkpoint. The ABEND code issued will be 117.

CKPTID Flag

The CKPTID flag contains the ID of the last checkpoint taken. It is a character field with a length of 8 bytes. It contains blanks prior to the first checkpoint.

COLUMN Flag

The COLUMN flag is set after the successful completion of an array operation. It is a fixed field with a length of 4 bytes. It contains a numeric value set to the column number being processed. If the operation fails, the flag is set to zero. If an array operation is suppressed, the COLUMN flag value is not changed.

COMMAND Flag[G]

The COMMAND flag contains the command identifier. It is a character field with a length of 8 bytes. Flag values are:

Input:	Sequential or Serial:	GETFIRST	Get first segment within a parent.
		GETNEXT	Get the next occurrence of the segment type previously obtained.
	Key driven: (transaction driven,[4] ICF, start search)	GETFKEY	Get first segment with key greater than or equal to the supplied value.
		GETKEY	Get the segment with the key provided.
Output:		REPLACE[4]	Replace the segment. Some data has been changed.
		ADD[4]	Add the segment. A new segment has been created.
		NOCHANGE[4]	This segment has not changed (however, BDAM may still need to see it).
		DELETE[4]	Delete the segment. This segment was explicitly deleted by the application by a transaction or DELETE flag setting. The parent segment has not been deleted.
		INIT	Initialization request call.
		TERM	Termination request call.

The COMMAND flag settings are not strictly necessary if LM statements are used. Some mapping developers may prefer to write a single request to handle all situations. In this case, the flag becomes valuable to identify the activity required.

The information is primarily useful for directing the operation of the mapping request, but may also be useful to a database manager. The flags are initialized prior to entering a mapping request.

CONDCODE Flag

The CONDCODE flag allows you to communicate with your job control. It is a binary field with a length of 2 bytes. The flag has an external effect only on OS/390, CMS, and VSE.

Note: It is your responsibility to ensure that the final value of the condition code is recognized by the operating system.

You can access this flag during processing for arithmetic calculations, selection, and output. It is initialized to zero at the beginning of the run. During processing, it contains only the values supplied by you. The values normally supplied by VISION:Builder are not available during request processing.

VSE: Placing any non-zero numeric value in the CONDCODE flag cancels the job due to program control at the end of the current step. Thus, you can cancel a job by replacing 1 into CONDCODE and setting the EOF flag to all Es.

OS/390 and CMS: At the end of the run, after all request processing is completed and all files have reached end of file, the values you placed in the CONDCODE flag during request processing are added to the condition code values normally supplied by VISION:Builder.

- This procedure allows you to control the range of condition codes without losing the settings provided by VISION:Builder.
- Only the last value of CONDCODE is used; intermediate values have no effect.
- If the CONDCODE flag is invalid at the end of the run,VISION:Builder sets it to 20 before adding it to the normal condition code value. In CMS, the condition code is referred to as the return code.

CSTATUS Flag

The CALL status flag (CSTATUS flag) indicates the status of a CALL. It is a character field with a length of 4 bytes. The CSTATUS flag values are set by VISION:Builder and indicate whether a CALL was suppressed by VISION:Builder and why.

- A CALL is suppressed if a parameter specifies an invalid or missing field. The following table shows the settings of the CSTATUS flag:

Value	Meaning
PMIS	Parameter missing.
PINV	Parameter invalid.
blank	CALL successful.

- The CSTATUS flag should be examined before using a value from the RETURNCD flag because a suppressed call would not allow the called routine to set the RETURNCD flag.

DATE Flag

The DATE flag records the date, acquired from the operating system, in the format: MMM DD, YYYY, where MMM equals 3 alpha characters for month, DD equals 2 numerals for day of month, and YYYY equals 4 numerals for year (that is, January 15, 1995 equals JAN 15, 1995). It is a character field with a length of 12 bytes.

DELETE Flag❹

The DELETE flag directs VISION:Builder to delete records or segment occurrences from a master file or to force rejection of a transaction record during processing. It is a fixed field with a length of 1 byte. The settings for the DELETE flag are shown in the following table.

Setting	Result
0	All M4OLD data is output to M4NEW. The DELETE flag is automatically set to zero each time a record is processed.
1	The master file record is deleted before it can be written to the new master file following standard request processing. Records can also be dynamically deleted from the master file in type M, 2, and 3 procedures/requests. Deleted records can be output to an audit file.
2	The appropriately marked lower level segments are deleted from the new master record before it is written to the new master file. Segments are marked for deletion procedurally by placing blank or zero into the segment key and placing 2 into the DELETE flag. Segments can be marked for deletion in type N, M, 2, and 3 procedures/requests. This is not supported when processing a relational file.
4	The transaction is rejected without the master file being updated. This occurs only in type 1 and 2 procedures/requests.

ECORD Flag

The ECORD flag indicates the status of the coordinated files in relation to the master file or coordinated file to which it is chained. It is a character field with a length of 9 bytes. The values for the ECORD flag are shown in the following table.

Value	Meaning
M	The coordinated file and master file keys are equal. The coordinated file has a match, and its record is available for processing.
X	The coordinated file is high in relation to the master file key. The coordinated file record is not available for processing. This value is also used when the coordinated file reaches the end of file, when no coordinated file exists for the run, or if the file is request read.
L	The coordinating file key is less than the master file key. The record is available for processing.

In the ECORD flag, each character indicates the current status of a corresponding coordinated file. Partial fielding is used to determine the status of one or more of the coordinated files, as shown below.

File Name	Partial Field Start and Length
Coordinated File 1	1,1
Coordinated File 2	2,1
Coordinated File 3	3,1
Coordinated File 4	4,1
Coordinated File 5	5,1
Coordinated File 6	6,1
Coordinated File 7	7,1
Coordinated File 8	8,1
Coordinated File 9	9,1

The ECORD flag is initialized whenever a master file record is read and/or created. For each coordinated file specified on an RF statement, the ECORD flag is initialized to X. As each coordinated file record is advanced, its status (in relation to the master file record being matched) is reflected in the ECORD flag. The ECORD flag indicates file status and must not be altered. Altering the ECORD flag does not change the physical status of the files but misleads subsequent processing.

EOF Flag

The EOF flag allows you to detect or force an end of file on sequentially read input files. It is a character field with a length of 11 bytes. It can be tested to determine if an input file exists. Alternatively, it can be used to terminate the reading of an input file by storing the appropriate value in the flag.

Note: When forcing end of file, if any value except E is placed into the EOF flag, the results are unpredictable.

Each character pertains to an input file as shown in the following chart and can have one of three values:

Value	Meaning
E	File has reached end of file.
Y	File has not reached end of file.
N	File does not exist.

Input File Name	Partial Field Specification
M4OLD	1,1
M4TRAN	2,1
M4CORD1	3,1
M4CORD2	4,1
M4CORD3	5,1
M4CORD4	6,1
M4CORD5	7,1
M4CORD6	8,1
M4CORD7	9,1
M4CORD8	10,1
M4CORD9	11,1

FDNAME Flag

The FDNAME flag contains the file name as in columns [1-8] on the FD statement. It is a character field with a length of 8 bytes.

It is initialized prior to entering a mapping request. The information in the flag is generally required by database managers.

FILE Flag

The FILE flag contains the VISION:Builder logical file name. It is a character field with a length of 8 bytes. Values are: M4OLD, M4NEW, and M4CORD1–9.

The information is primarily useful for directing the operation of the mapping request, but can also be useful to a database manager.

FILEID Flag[G]

The FILEID flag contains the file identification as provided in columns [11-18] of the FD statement. It is a character field with a length of 8 bytes.

It is initialized prior to entering a mapping request. The information in the flag is generally required by database managers.

ISDATE Flag

The ISDATE flag records the system date in two International Standards Organization (ISO) formats.

Normal processing/standard reporting (8 characters)

This flag records the date in the format of YYYYMMDD, where YYYY equals 4 numerals for year, MM equals 2 numerals for month, and DD equals 2 numerals for day (that is, January 15, 1995 equals 19950115).

Formatted reporting (10 characters)

This flag records the date in the format of YYYY-MM-DD (1995-01-15).

Note: The date delimiter (-) is an installation option and can be changed using M4PARAMS. See the *Installation Guide*.

JULANX Flag

The JULANX flag records the system date in two formats.

Normal processing/reporting (7 characters)

This flag records the date in the format of YYYYDDD, where YYYY equals year and DDD equals day in the year (that is, January 15, 1995 equals 1995015).

Formatted reporting (8 characters)

This flag records the date in the format of YYYY.DDD (1995.015).

Note: The date delimiter (.) is an installation option and can be changed using M4PARAMS. See the *Installation Guide*.

JULIAN Flag

The JULIAN flag records the system date in two formats.

Normal processing/reporting (5 characters)

This flag records the date in the format of YYDDD, where YY equals year and DDD equals day in the year (that is, January 15, 1995 equals 95015).

Formatted reporting (6 characters)

This flag records the date in the format of YY.DDD (95.015).

Note: The date delimiter (.) is an installation option and can be changed using M4PARAMS. See the *Installation Guide*.

LILIAN Flag

The LILIAN flag contains the current date in Lilian date format as a fixed-binary number with a length of 4 bytes. The valid range of Lilian dates is 1 – 3,074,324 (October 15, 1582 to December 31, 9999). VISION:Builder automatically converts a LILIAN flag field into character format using M4PARAMS settings when a LILIAN flag field is specified for printing. The format is identical to the TODAYX flag conventions unless an override edit picture is specified.

LNUMBER FLAG

The LNUMBER flag is a text processing partial field flag that specifies the number of characters in the left part of the field. It is a fixed field with a length of 4 bytes. It is used in a scan left or right operation and can be tested, modified, or used in computation during request processing.

- The value LN can be used in place of the standard partial field specification in order to access a particular area of the scanned field.
- The value LN (dynamic partial field specification) cannot be used on the output specification (Rn) statement, nor in place of the flag names in operand A, operand B, or result specifications. For example, if LN is entered in starting character on the PR statement, the current value of LNUMBER is used at the time the PR statement is executed.

Because the value of LNUMBER may change during processing, you can use this flag to perform dynamic partial field operations.

A dynamic partial field specification might be in error during processing because:

- The partial field flag is invalid.
- The partial field specification is outside the current size of the field.
- The starting character or number of characters value is less than 1.

If the dynamic partial field specification is in error, the field to which it is applied becomes invalid for the current operation. If an invalid dynamic partial field is applied to a result field, the result field becomes invalid (except in a scan and substitute (SS) operation, where the result field is invalid for the current operation only).

LSTART Flag

The LSTART flag is a text processing partial field flag that specifies the starting position of the left part of the field. It is a fixed field with a length of 4 bytes. It is used in a scan left or right operation and can be tested, modified, or used in computation during request processing.

- The value LS can be used in place of the standard partial field specification on the processing and record selection statement in order to access a particular area of the scanned field.
- The value LS (dynamic partial field specification) cannot be used on the output specification (Rn) statement, nor in place of the flag names in operand A, operand B, or result specifications. For example, if LS is entered in starting character on the PR statement, the current value of LSTART is used at the time the PR statement is executed.

Because the value of LSTART may change during processing, you can use this flag to perform dynamic partial field operations.

A dynamic partial field specification might be in error during processing because:

- The partial field flag is invalid.
- The partial field specification is outside the current size of the field.
- The starting character or number of characters value is less than 1.

If the dynamic partial field specification is in error, the field to which it is applied becomes invalid for the current operation. If an invalid dynamic partial field is applied to a result field, the result field becomes invalid (except in a scan and substitute (SS) operation where the result field is invalid for the current operation only).

LSTATUS Flag

The LSTATUS flag indicates the status code following the execution of any segment operation. It is a character field with a length of 4 bytes. The status information can be examined to determine if the operation was successful, failed, or suppressed. This flag and its contents are independent of the type of file being accessed (for example, IMS, sequential, ISAM).

As part of the segment operation, there is an implied compare to blanks in the LSTATUS flag when there is an NS or GS operation immediately following a segment operation. The NS or GS branch will be taken on any status other than blanks. At the "branch-to" location, the status information can be examined to determine why the operation failed or was suppressed. This can be useful in debugging applications or in determining sources of erroneous input to applications.

The LSTATUS codes and explanations are shown in the following table. The priority column indicates the order in which the LSTATUS flag settings occur.

	Applicable To				
LSTATUS Value	**FS**	**FF FL**	**RS**	**Priority**	**Explanation**
␣␣␣␣	Yes	Yes	Yes		The operation executed successfully.
NREC	Yes	Yes	No	1	The operation is suppressed. The current record is unavailable for the referenced segment.
AUTO	Yes	Yes	Yes	2	The operation is suppressed. The segment type referenced is already in an automatic loop; or a field in operand B is part of the segment type referenced, or part of an unreferenced (by an FS or RS operator) dependent segment.
TRAN❹	Yes	Yes	Yes	3	The operation is suppressed. The segment type referenced was matched, created, or inserted by the current transaction record. It can only be set in the type M procedure or its subroutines.
TDEL❹	Yes	Yes	Yes	3	The operation is suppressed. The segment type referenced was deleted by the current transaction record. It can only be set in the type M procedure or its subroutines.

LSTATUS Value	Applicable To FS	FF FL	RS	Priority	Explanation
INHR	Yes	Yes	Yes	4	The operation is suppressed. The segment type referenced in a subroutine procedure is in an automatic loop inherited from the calling procedure.
BMIS	Yes	No	No	6	A field in the operand B has a missing value.
BINV	Yes	No	No	6	A field in the operand B has an invalid value.
NPAR	Yes	Yes	Yes	7	The operation is suppressed. The segment type referenced has a parent segment that is unavailable or does not exist.
NFND	Yes	Yes	No	8	The operation fails. VISION:Builder cannot find any segment occurrences that satisfy the operation.
NOMO	Yes	No	No	8	The operation fails. The next occurrence of a referenced segment type does not exist.

M4AUDIT Flag❹

The M4AUDIT flag provides access to the number of records output to the M4AUDIT file during application execution. It is a fixed field with a length of 4 bytes. The M4AUDIT flag can be referenced in any type of request (except preselection); however, it can only be referenced if the M4AUDIT file is being used in the application.

M4CORDn (n=1 to 9) Flags

There are separate flags provided for each one of the nine VISION:Builder coordinated files, M4CORDn where n=1 to 9. These are fixed fields with lengths of 4 bytes.

These flags (M4CORD1, M4CORD2, M4CORD3, M4CORD4, M4CORD5, M4CORD6, M4CORD7, M4CORD8, and M4CORD9) provide access to the number of records read from each of the various coordinated files during application execution. These coordinated file flags can be referenced in any type of request (except preselection); however, they can only be referenced if the coordinated file they reference is being used in the application.

M4NEW Flag❹

The M4NEW flag provides access to the number of records output to the M4NEW file during application execution. It is a fixed field with a length of 4 bytes. The M4NEW flag can be referenced in any type of request (except preselection); however, it can only be referenced if the M4NEW file is being used in the application.

M4OLD Flag

The M4OLD flag provides access to the number of records read from the M4OLD file during application execution. It is a fixed field with a length of 4 bytes. The M4OLD flag can be referenced in any type of request (except preselection); however, the flag can only be referenced if the M4OLD file is being used in the application. When using update-in-place, the flag will represent only the records read from M4OLD.

M4REJECT Flag❹

The M4REJECT flag provides access to the number of records output to the M4REJCT file during application execution. It is a fixed field with a length of 4 bytes. The M4REJECT flag can be referenced in any type of request (except preselection); however, it can only be referenced if the M4REJCT file is being used in the application.

M4SUBFn (n= 0 to 9) Flags

There are separate flags provided for each one of the ten VISION:Builder subfiles, M4SUBFn where n=0 to 9. These are fixed fields with lengths of 4 bytes.

These flags (M4SUBF0, M4SUBF1, M4SUBF2, M4SUBF3, M4SUBF4, M4SUBF5, M4SUBF6, M4SUBF7, M4SUBF8, and M4SUBF9) provide access to the number of records output to each of the various subfiles during application execution. These subfile flags can be referenced in any type of request (except preselection); however, they can only be referenced if the subfile they reference is being used in the application.

M4TRAN Flag❹

The M4TRAN flag provides access to the number of records read from the M4TRAN file during application execution. It is a fixed field with a length of 4 bytes. The M4TRAN flag can be referenced in any type of request (except preselection); however, it can only be referenced if the M4TRAN file is being used in the application.

MISSPASS Flag

The MISSPASS flag value is automatically set by VISION:Builder only at the start of the all-miss pass (ECORD setting of all Xs). It is a character field with a length of 1 byte. This flag will contain either an X or an M during the all-miss pass. An X setting indicates that no hits occurred during coordination for the master record; an M setting indicates at least one hit occurred during coordination. The MISSPASS flag is blank when the all-miss pass is not in progress.

While the ECORD flag indicates the status of coordinated files, the MISSPASS flag indicates the status of the master file. The MISSPASS flag should not be used with RPCORDONLY because its value during a match or low cycle is blank and meaningless.

MNUMBER Flag

The MNUMBER flag is a text processing partial field flag that specifies the number of characters in the middle (matching) part of the field. It is a fixed field with a length of 4 bytes. It is used in a scan left or right operation and can be tested, modified, or used in computation during request processing.

- The value MN can be used in place of the standard partial field specification in order to access a particular area of the scanned field.
- The value MN (dynamic partial field specification) cannot be used on the output specification (Rn) statement, nor in place of the flag names in operand A, operand B, or result specifications. For example, if MN is entered in starting character on the PR statement, the current value of MNUMBER is used at the time the PR statement is executed.

Because the value of MNUMBER may change during processing, you can use this flag to perform dynamic partial field operations.

A dynamic partial field specification might be in error during processing because:

- The partial field flag is invalid.
- The partial field specification is outside the current size of the field.
- The starting character or number of characters value is less than 1.

If the dynamic partial field specification is in error, the field to which it is applied becomes invalid for the current operation. If an invalid dynamic partial field is applied to a result field, the result field becomes invalid (except in a scan and substitute (SS) operation where the result field is invalid for the current operation only).

MODE Flag[G]

The MODE flag contains information about the application mode of operation. The following table shows the X and Y values. It is a character field with a length of 2 bytes.

In Operation Mode X =		In Update Mode Y =	
S	Standard Processing	R	Retrieval Update
M	MOSAIC Processing	U	Update
		L	Load Module

It is initialized prior to entering a mapping request. The information in the flag is generally required by database managers.

MSTART Flag

The MSTART flag is a text processing partial field flag that specifies the starting position of the middle (matching) part of the field. It is a fixed field with a length of 4 bytes. It is used in a scan left or right operation and can be tested, modified, or used in computation during request processing.

- The value MS can be used in place of the standard partial field specification on the processing and record selection statement in order to access a particular area of the scanned field.
- The value MS (dynamic partial field specification) cannot be used on the output specification (Rn) statement, nor in place of the flag names in operand A, operand B, or result specifications. For example, if MS is entered in starting character on the PR statement, the current value of MSTART is used at the time the PR statement is executed.

Because the value of MSTART may change during processing, you can use this flag to perform dynamic partial field operations.

A dynamic partial field specification might be in error during processing because:

- The partial field flag is invalid.
- The partial field specification is outside the current size of the field.
- The starting character or number of characters value is less than 1.

If the dynamic partial field specification is in error, the field to which it is applied becomes invalid for the current operation. If an invalid dynamic partial field is applied to a result field, the result field becomes invalid (except in a scan and substitute (SS) operation where the result field is invalid for the current operation only).

MSTATUS Flag

The MSTATUS flag is used by the mapping request to instruct VISION:Builder on a specific action to take subsequent to completion of the mapping request. It is a character field with a length of 6 bytes.

MSTATUS is initialized to blanks prior to each call to a mapping request. Permissible return values are listed in the following table.

Value	Explanation
blank	Mapping successfully completed.
NFOUND	Segment not found. Useful for input mapping only. Signifies to VISION:Builder that there was no data to fill the skeleton segment, either because of having run out of repeated segments, or because of failure to locate a keyed segment. In the process of building a hierarchical record (standard or MOSAIC processing, non-keyed operations), VISION:Builder returns to the mapping requests for each segment type to request repeated segments. Thus, the mapping requests must be programmed to issue NFOUND for each segment type to terminate a series of repeated segments.
STOPnn	Stop the run and set the condition code, which is supported in OS/390 only, to nn. Used by mapping requests to cease processing.

The information is primarily useful for directing the operation of the mapping request, but can also be useful to a database manager.

OWN Flag

The OWN flag is used by own-code routines to communicate between own-code requests. It is a character field with a length of 16 bytes.

It is not examined or altered by VISION:Builder in any way. The flag name OWN can be entered in field name A, B, or C depending on the operation being performed.

PAGE Flag

The PAGE flag is used only in formatted reporting and specifies the placement of the page numbers, which are printed left-aligned in the title and summary lines of formatted reports. It is a character field with a length of 6 bytes. (Enter the PAGE flag as F.PAGE.)

PASSWORD Flag[G]

The PASSWORD flag contains the password as provided on the RF statement for the appropriate VISION:Builder file. It is a character field with a length of 8 bytes. It is available for all mapping requests but is expected to be most valuable to the initialization mapping request to control access to the database. This field is set prior to a mapping request receiving control. Valid only in mapping requests.

RESTART Flag[I]

The RESTART flag is used by requests to determine the restart status of a VISION:Builder run. It is a character field with a length of 8 bytes. If a run has been restarted, the RESTART flag contains the checkpoint ID at which the restart occurred. Otherwise, the value of the RESTART flag is all blanks.

RETURNCD Flag

The return code (RETURNCD flag) reflects the status of a CALL. It is a fixed field with a length of 4 bytes. The values are set by the called routine and the contents can be examined after a CALL. If the flag is included in a report, VISION:Builder treats the flag as a 2-byte fixed point field (output width of 7, maximum printable value 99,999). The value in this field is the value in general register 15 upon return to VISION:Builder from a called routine.

RNUMBER Flag

The RNUMBER flag is a text processing partial field flag that specifies the number of characters in the right part of the field. It is a fixed field with a length of 4 bytes. It is used in a scan left or right operation and can be tested, modified, or used in computation during request processing.

- The value RN can be used in place of the standard partial field specification in order to access a particular area of the scanned field.
- The value RN (dynamic partial field specification) cannot be used on the output specification (Rn) statement, nor in place of the flag names in operand A, operand B, or result specifications. For example, if RN is entered in starting character on the PR statement, the current value of RNUMBER is used at the time the PR statement is executed.

Because the value of RNUMBER may change during processing, you can use this flag to perform dynamic partial field operations.

A dynamic partial field specification might be in error during processing because:

- The partial field flag is invalid.
- The partial field specification is outside the current size of the field.
- The starting character or number of characters value is less than 1.

If the dynamic partial field specification is in error, the field to which it is applied becomes invalid for the current operation. If an invalid dynamic partial field is applied to a result field, the result field becomes invalid (except in a scan and substitute (SS) operation where the result field is invalid for the current operation only).

ROW Flag

The ROW flag is set after successfully completing an array operation. It is a fixed field with a length of 4 bytes. It contains a numeric value set to the ROW number being processed. If the operation fails, the flag is set to zero. If an array operation is suppressed, the ROW flag is not suppressed.

RSTART Flag

The RSTART flag is a text processing partial field flag that specifies the starting position of the right part of the field. It is a fixed field with a length of 4 bytes. It is used in a scan left or right operation and can be tested, modified, or used in computation during request processing.

- The value RS can be used in place of the standard partial field specification on the processing and record selection statement in order to access a particular area of the scanned field.
- The value RS (dynamic partial field specification) cannot be used on the output specification (Rn) statement, nor in place of the flag names in operand A, operand B, or result specifications. For example, if RS is entered in starting character on the PR statement, the current value of RSTART is used at the time the PR statement is executed.

Because the value of RSTART may change during processing, you can use this flag to perform dynamic partial field operations.

A dynamic partial field specification might be in error during processing because:

- The partial field flag is invalid.
- The partial field specification is outside the current size of the field.
- The starting character or number of characters value is less than 1.

If the dynamic partial field specification is in error, the field to which it is applied becomes invalid for the current operation. If an invalid dynamic partial field is applied to a result field, the result field becomes invalid (except in a scan and substitute (SS) operation where the result field is invalid for the current operation only).

RSTATUS Flag

The RSTATUS flag indicates the results of the most recent request-read (RD, RE, or RG) operation performed in a PR statement. It is a character field with a length of 4 bytes.

The RSTATUS flag is set whenever a read (RD), read equal key (RE), or read greater than or equal key (RG) operator is used with coordinated files. The settings of the RSTATUS flag are shown in the following table.

Value	Explanation
AINV	Operand A contains an invalid field. The RE or RG operation was not performed.
AMIS	Operand A was missing. The RE or RG operation was not performed.
KEQL	The RE or RG operation was performed and a root segment with a key equal to the key in operand A was successfully returned, or an RD was successful.
KGTR	The RG operation was performed and a root segment with a key greater than the key in operand A was successfully returned.
NREC	The operation was performed but no record was found.
RIGN	The read operation was ignored. A segment in the specified coordinated file is in a loop or under control of a find segment (FS) operation.
EGRP	Specifies the end of a group. A sequential RD operator following a direct-read operator compares the key of each root segment retrieved to the search argument for the direct-read. When the key no longer matches, the RSTATUS flag indicates the end of group. EGRP is also set if end of file is reached on a sequential RD operation. In this situation, the EOF flag for the appropriate coordinated file is set to E and reset to Y on the next RE or RG operation for that file. This is the only situation in which VISION:Builder resets an EOF flag value. Successive RD operations without any intervening RE or RG operations continue to return the EGRP and EOF flag values.

An NS or GS statement immediately following an RE or RG operator causes an implied compare of the RSTATUS flag.

- If the RE or RG operation fails (that is, the RSTATUS flag is equal to AINV, AMIS, or NREC), the NS branch is taken.
- If the RE or RG operation is successful (that is, the RSTATUS flag is equal to KEQL, KGTR, or RIGN), processing continues with the next instruction.

The GS statement can be used in the same way as the NS statement to determine success or failure.

SEGNAME Flag[G]

The SEGNAME field contains the segment name as defined in columns [11-18] of the LS statement. It is a character field with a length of 8 bytes. If an alias segment name is provided on an LB statement in a run data group, the SEGNAME flag still refers to the segment name as defined on the LS statement.

The information is primarily useful for directing the operation of the mapping request, but can also be useful to a database manager. It is initialized prior to entering a mapping request.

SQL Flag[D]

The SQL flag contains status information after an attempt is made to insert a row into a DB2 table with a UNIQUE index. It is a character field with a length of 4 bytes. It contains the character string DUPL if a matching row already exists in the table. It contains blanks if a matching row does not exist.

- If F.SQL equals blanks, the record is successfully inserted.
- If F.SQL equals DUPL and Scan/Terminate does not equal 4, the run terminates; however, if Scan/Terminate equals 4, processing continues (the rejected row can be output to a report or another subfile). Note that this is the only instance in which the Scan/Terminate value determines processing based on message types other than type 4.
- If type 0 and type 1 messages are suppressed, there is no indication that the attempt to insert the row failed.

SSCOUNT Flag

The SSCOUNT flag counts the number of matches found during the REPLACE operation. It is a fixed field with a length of 2 bytes. The flag is set to zero at the start of each run. The settings are shown in the following table.

Setting	Result
0	There are no successful substitutes or the search-string is longer than the field to be modified.
-1	Used with variable length fields. If a REPLACE operation causes the maximum length of a type V field to be exceeded, the operation does not take place, and the field to be modified is made invalid.
-2	The field to be modified or search-string is null and/or invalid, and/or the replace-string is invalid prior to a REPLACE operation.

STRAN Flag❹

The STRAN flag interrogates the status of the current transaction at each segment level and indicates the actions applied to the master file record by the current transaction. It is a character field with a length of 9 bytes. The STRAN flag is used in type M requests and their subroutines but is available to any request type.

Each of the 9 bytes of the STRAN flag corresponds to a segment level in a structured record. The settings are shown in the following table.

Setting	Explanation
Blank	No action at this level.
M	Match actions only occurred at this level. All match fields for the segment at this level were matched (transaction action code M), but no fields in the segment were updated (action codes B, R, A, S, P, or X).
I	A segment was inserted at this level. This setting applies to level 1 (root) as well as lower levels 2 through 9. Thus, action code C or I was used or M with default create or insert specified. In addition, update actions (codes B, R, A, S, P, or X) may have been applied after the create or insert.
U	Update actions (codes B, R, A, S, P, or X) were applied to the segment at this level, but C or I was not specified.
D	The segment at this level was deleted by a transaction (action code E). This setting applies only to the level that was explicitly deleted by the transaction, not to lower level dependent segments. This setting is not apparent at level 1 because deleted master records are not presented to any type of request.

The STRAN flag is reset to blanks before obtaining the next logical transaction record (that is, either reading a new transaction record or retrying the last one). However, the STRAN flag is not revalidated if you set it to an invalid value. This is consistent with the treatment of other user flags.

The STRAN flag is set to non-blank values as the transaction actions are successfully applied.

If a transaction is referenced in the type 4 request and rejected because of update errors, the STRAN flag reflects the match actions that occurred but does not reflect the insert, update, or delete actions. These actions were either not yet applied or were backed out before the type 4 request received control.

TIME Flag

The TIME flag records the time of day at which the run was started. It is a character field with a length of 8 bytes. The TIME flag prints as HH.MM.SS, where HH equals hours, MM equals minutes, and SS equals seconds. Leading zeroes are added where necessary.

TODAY Flag

The TODAY flag records the system date in two formats.

Normal processing/standard reporting (6 characters)

The TODAY flag records the date in the format MMDDYY, where MM equals 2 numerals for month, DD equals 2 numerals for day, and YY equals 2 numerals for year (that is, January 15, 1995 equals 011595).

Formatted reporting (8 characters)

The TODAY flag records the date in the format MM/DD/YY (01/15/95).

Note: The order of month, day, and year and the date delimiter (/) are installation options and can be changed using M4PARAMS. See the *Installation Guide*.

TODAYX Flag

The TODAYX flag records the system date in two formats.

Normal processing/standard reporting (8 characters)

The TODAYX flag records the date in the format MMDDYYYY, where MM equals 2 numerals for month, DD equals 2 numerals for day, and YYYY equals 4 numerals for year (that is, January 15, 1995 equals 01151995).

Formatted reporting (10 characters)

The TODAYX flag records the date in the format MM/DD/YYYY (01/15/1995).

Note: The order of month, day, and year and the date delimiter (/) are installation options and can be changed using M4PARAMS. See the *Installation Guide*.

TRAN Flag[4]

The TRAN flag indicates the maintenance status of a master file record and/or the rejection of a transaction record. It is a fixed field with a length of 1 byte. The settings are shown in the following table.

Value	Conditions
0	No transaction records exist for the master file record.
1	The master file record was updated or matched by a transaction record.
2	The master file record was created by a transaction record.
3	Conditions for values 1 and 2 both apply.
4	One or more transaction records against this master file record were rejected.
5	Conditions for values 1 and 4 both apply.
6	Conditions for values 2 and 4 both apply.
7	Conditions for values 1, 2, and 4 apply.
8 or higher	Invalid.

XTRAN Flag❹

The XTRAN flag identifies the reasons for rejected transactions. It is a fixed field with a length of 1 byte. When more than one error is detected in a transaction, only the last one is described.

The XTRAN settings are shown in the following table.

Value	Explanation
0	Inactive setting.
1	Request rejection.
2	The transaction record does not match any of the defined identifiers.
3	The field in the transaction is outside the maximum or minimum values specified in the transaction definition.
4	A transaction record key is less than the preceding transaction record key; the transaction file is out of sequence.
5	Transaction record key field with M, D, or C action cannot be converted for matching purposes.
6	The contents of the date field in the transaction do not meet the date validation criteria.
7	The field in the transaction record does not conform to the input edit pattern specified in the transaction definition.
8	The transaction record key does not match any master file key and a create is not specified.
9	A create action has been specified for a record that already exists on the master file.
10	Field associated with an M, E, C, D, or I action cannot be converted for matching purposes or a field associated with B, R, A, S, P, or X action causes an arithmetic or conversion error.
11	A non-record key field (that is, any field other than a record key field) associated with an M action cannot be located in the master file record.
12	The segment for which a delete segment action (E) is specified does not exist in the master file record.
13	The insert action (I) has been specified for a segment that is repeated a fixed number of times. There are no “empty” segments into which the insert can be made.

Value	Explanation
14	A segment to be inserted already exists in the master file record.
15	This value occurs with variable length records only. The requested insertion would cause the maximum record length to be exceeded.
17	The count field controlling the segment to be inserted is too small to contain the number of subordinate segments being inserted.

Chapter

4

VISION:Builder Technical Notes

Statement Input Sequence

With few exceptions, there is no required order for VISION:Builder input statements. However, some statement types are required for some runs and are explained in this chapter. In every VISION:Builder run, except a Utility Control run, the first statement of M4INPUT must be an RC statement. In general, it is necessary that all statements with the same name in columns [1-8] be grouped together. You can determine the most efficient order simply by using the formatted source listings as a guide.

Definition Runs

A definition run has blanks after position 10 on the RC statement. In definition runs, the statement types that can be included are:

RP IT CT		TB TE	For each Table Set
FD LS L0 Ln Dn LX L A L M G LR D	For each File Set	TD 4 TL 4 DB RF LB RT D RP RG 4 OC CP	For each Transaction Set For each Run Data Group
A D B0 Bn B A A A	For each Array Set	PA RA WH D CR	

- CT statements can appear anywhere as long as they do not break up a set.
- IT statements can appear anywhere as long as they appear before the item they track and before any CT statements in the run for the item they track.
- TB and TE statements for a given name make up a table set.
- FD and Ln statements for a file name make up a file set.
- AD and Bn make up an array set and the TD 4 and TL 4 statements make up a transaction set.
- AA statements can appear anywhere in any set and will print on the source listings where they appear in M4INPUT.

The order of the sets is unimportant except that a table name appearing in a file or array set must be defined either in a table set or in the library before it is used. A file name appearing in a transaction set must be similarly defined prior to being referenced.[4]

- TB, FD, and AD statements are required only when initially cataloging or deleting a table, file, or array definition.
- L1 to L9, D1, D2, LX, and LA statements require a preceding L0 statement in either the file set or in the library from a previous run.
- Similarly, B1 to B9 and BA statements require a preceding B0 statement.
- TD statements always require preceding TL statements and all TD and TL statements for the same identifier must be contiguous in the input stream.[4]

Processing Runs

A processing run has an FD name (M4OLD or NEW) in positions [11-18] on the RC statement. In a processing run, the RC statement can be followed by the CP, OC, PA, RA, RF, RG,[4] RP, RT,[D] and WH[D] statements to form a group. The order of these statements in the group is immaterial but, when they are used, they must precede all other VISION:Builder source statements.

AA statements can appear in the run control group and print where they appear in M4INPUT. If used, CT statements must be next, followed by IT statements. The order of any other statements in a processing run is at the discretion of the user, as long as all statements with the same request name are contiguous.

The following is the order of appearance of request statements on a source listing and can be used as a guide:

ER TF PR En Jn Kn Pn Rn Fn Tn	Request Set

Not all particular statement types are required for any given request and default options will be provided for any statement type that is not included in a request. It should be noted that both a Tn and an Fn statement (for a given value of n) cannot be included for the same request name, except for formatted sectional reporting when the page-title section is not used. CR statements can replace any request set. AA statements can appear in any request set and will print where they appear in M4INPUT.

In a processing run, you can include file definition statements to update or create a file definition. These instream file definition statements, however, are not cataloged and do not permanently change any definition already cataloged. The definitions are temporary, lasting only for the duration of the processing run. No update to the library takes place. These statements should be placed immediately after the run control group.

Reporting Runs

A reporting run has only an S in position [25] on the RC statement. In a reporting run, the RC statement can be followed by the RP, CP, OC, and/or AA statements to form a group.

Processing File Definitions

The file definition process includes creation, overrides, updates, file name modification, file identification modification, and so on.

Create a New File Definition

VISION:Builder completes the processing of a file definition whenever a change occurs in the file name field of the input stream. Each statement of a file definition is checked for errors and printed in statement image format along with any system messages. After all statements and related messages are printed, a final message informs the user whether or not the file definition has been cataloged. The glossary listing concludes the processing of the file definition.

A definition is not cataloged if a system message with a severity code of 3 or greater is given. If cataloged, a glossary always prints unless N is specified for no glossary listing.

File Definition Overrides

The characteristics of cataloged file definitions can be temporarily overridden during the execution of a VISION:Builder processing run. The specifications details for run-time overrides are identical to those for updating a definition in a definition maintenance run (see Update an Existing File Definition on page 4-5) with one exception: deleting an entire file definition is not allowed (Y in position [19] of the FD statement), but replacing an entire file definition is allowed (R in position [19] on the FD statement).

Unlike a definition maintenance run, the effects of the overrides are temporary, lasting for the duration of the processing run; no update to the M4LIB takes place. If file definition override statements are included in a processing run, they must immediately follow the run control group.

File definitions can be overridden in a processing run only for master file definitions, coordinated file definitions, and file definitions for transactions.

The sequence of input to VISION:Builder is as follows:

FD
LS
L0
Ln
Dn
LX
LA
LM[G]
LR[D]

Update an Existing File Definition

Updates or changes can be made to an existing file definition without a complete redefinition of the file (except for mapped file definitions).

- The following sections detail typical areas of change. Each section describes the input required. It might be necessary to combine several of these changes, depending on what is required.
- Updates or changes to the cataloged file definition can be made during any definition maintenance run.
- Temporary file definition updates or changes that exist only for the duration of a run can be made during a processing run.
- If you include the modified file definition statements into the processing run, VISION:Builder temporarily overrides the cataloged file definition.

Change the File Name of an Existing File Definition

The file name can be changed only by reentering all file definition statements with the new file name. The file definition can be retrieved from the library by using source statement retrieval. The file definition containing the old file name can be deleted by entering an FD statement that deletes the existing file definition (see Delete an Existing File Definition on page 4-5).

The old definition can be deleted and the new definition entered in the same definition/maintenance run.

Delete an Existing File Definition

An existing file definition can be deleted by entering an FD statement containing the following information:

- Existing file name.
- VISION:Builder statement type FD.
- Delete specification.

The aggregate of the FD statement and all LM, LS, Ln, and LR statements defining a file is called the file definition.

A Y in the delete specification is not allowed during a processing run (instream file definitions).

Add or Change the File Identification

The file identification can be added to or changed only by entering an FD statement containing the following information:

- Existing file name.
- VISION:Builder statement type FD.
- New file identification.

Delete an Existing File Identification

The file identification can be deleted (changed to blank) only by deleting the entire file definition and redefining it. This can be done only in a definition run.

Print a Glossary from an Existing File Definition

A glossary can be printed from an existing file definition in the glossary specification format by entering an FD statement containing the following information:

- Existing file name.
- VISION:Builder statement type FD.
- Glossary.

This can be done in either a definition or processing run.

Change the Glossary

The glossary can be temporarily changed to be printed in a new format. This change is in effect only until the end of the definition/maintenance or processing run, at which time the glossary level reverts back to the original specification. To temporarily change the glossary level, enter an FD statement containing the following information:

- Existing file name.
- VISION:Builder statement type FD.
- Temporary glossary.

The glossary can be permanently changed from the existing specification by redefining the record format characteristics in a definition. Enter an FD statement containing the following information:

- Existing file name.
- VISION:Builder statement type FD.
- Existing file identification (if applicable).
- New glossary format specification.
- Existing record format specification.
- Existing record size (if applicable).
- Existing records per block specification (if applicable).
- Existing buffer size (if applicable).

Change the Record Format Characteristics

The record format characteristics are the aggregate of record format, record size, records per block, and buffer size. If these fields are to be changed, the complete FD portion of the file definition must be redefined. This can be done permanently in a definition run or temporarily in a processing run. Enter an FD statement containing the following information:

- Existing file name.
- VISION:Builder statement type FD.
- Existing or new record format specification.
- Existing or new record size (if applicable).
- Existing or new records per block specification (if applicable).
- Existing or new buffer size (if applicable).

To change the record format to type S from another format, be aware of the following additional considerations:

- LR statements must be included.
- DB2/SQL external name must be provided on the LS statement.
- If DL/I alias fields existed in the original definition, it is possible they will not conform to the syntax requirements of SQL. If that is the case, the update fails and a type 3 message is issued.

To change the record format from type S to another format, be aware of the following:

- LR statements are not allowed.
- In IMS definitions, the 18-character DB2/SQL external names are interpreted as DL/I aliases, using only the first 8 characters. No message is issued.

Delete a Field from an Existing File Definition

A field can be deleted from an existing file definition by entering an L0 statement containing the following information:

- Existing file name.
- VISION:Builder statement type L0.
- Existing field name.
- Delete specification.

Note: The deletion of a field from the file definition includes any existing lines of column heading text for that field.

Add a Field to an Existing File Definition

A field can be added to an existing file definition by entering an L0 statement containing the following information:

- Existing file name.
- VISION:Builder statement type L0.
- New field name.
- Segment number (if default option is not wanted).
- Level number (if default option is not wanted).
- Field location (if applicable).
- Field length.
- Field type (if default option is not wanted).
- Segment key (if applicable).
- Decimal places (if default option is not wanted).
- Count field for segment number (if applicable).
- Segment occurs n times (if applicable).
- Output edit code (if default option is not wanted).
- Output edit length (if default option is not wanted).

- DL/I alias name (if applicable).
- SQL column name (if applicable).

Add a Line of Column Heading Text

A line of column heading text can be added by entering an appropriate Ln statement (L1 to L9) containing the following information:

- Existing file name.
- Applicable VISION:Builder statement type L1 to L9.
- Existing field name.
- New column heading text (this field can be left blank if a blank line of column heading text is wanted).

Delete a Line of Column Heading Text

A line of column heading text can be deleted by entering an appropriate Ln statement (L1 to L9) containing the following information:

- Existing file name.
- Applicable VISION:Builder statement type L1 to L9.
- Existing field name.
- Delete specification.

Change the Text in a Line of Column Heading

A line of column heading text can be changed by deleting the existing line of text and adding the new line of column heading text to the file definition. (See Add a Line of Column Heading Text on page 4-9 and Delete a Line of Column Heading Text on page 4-9.)

Change Characteristics of an Existing Field

A change to any of the existing field characteristics can only be accomplished by deleting the existing field and adding the redefined field to the file definition.

The lines of column heading text can be deleted or changed without redefining the applicable field. If, however, any other changes are required to the characteristics of any existing field, the field must be deleted. When a field is deleted, the existing lines of column heading text are automatically deleted, making it necessary to also redefine the lines of column heading text. See Delete a Field from an Existing File Definition on page 4-8, Add a Field to an Existing File Definition on page 4-8, and Add a Line of Column Heading Text on page 4-9.

Change the DB2/SQL External Name

To change the table name (DB2/SQL external name) on the LS statement, you must delete the existing LS statement and provide a new LS statement with the following information:

- Existing file name.
- VISION:Builder statement type LS.
- Existing segment name.
- Existing segment number.
- Existing segment order.
- New DB2/SQL external name.

Add an LR Statement to an Existing File Definition

A logical relationship statement (LR statement) can be added to an existing file definition by entering an LR statement with the following information:

- Existing file name.
- Segment name.
- Logic level number (if applicable).
- Relational field name.
- Qualifier.
- VISION:Builder statement type LR.
- Sequence number.
- Connector (if applicable).
- Operation.
- Constant or primary key.

Delete an LR Statement

The following information must be specified when deleting an LR statement:

- File name.
- Segment name.
- Sequence number.
- VISION:Builder statement type LR.
- Delete specification.

Change an LR Statement

A logical relationship statement can be changed by deleting the existing LR statement and adding the new LR statement to the definition. Refer to the sections above for information on deleting and adding LR statements.

Output Edit Length

This technical note describes the output edit length rules for the different field types under the output edit specification of the file definition worksheet. The following table illustrates output edit specifications and results.

Code	Column	Output Edit Length	Printed Results
Blank	FLOATING	To allow the printing of a leading minus sign for negative values, add 1 position.	When no output edit codes are specified, a leading minus sign is printed if the value of the field is negative; positive values print unsigned.
$	FLOATING	To allow it to print, add 1 position.	A floating currency symbol inserts a leading dollar sign before the first value when any control break occurs and when summaries are taken. A decimal point always prints when one has been indicated by an entry in the no. of decimal places specification. A zero value in a field where a decimal place entry has been made prints as a decimal point followed by as many zeroes as there are decimal places. For a dollar and cents column, $.00 prints.
+	FLOATING	Add 1 position.	Prints leading sign (+ or -) depending on positive or negative value.
+	TRAILING	Add 1 position.	Prints trailing sign (+ or -) depending on positive or negative value.
-	FLOATING	Add 1 position.	Prints a leading minus sign if the value of the field is a negative number; positive numbers print unsigned (Default specification).
-	TRAILING	Add 1 position.	Prints a trailing minus sign if the value of the field is a negative number; positive numbers print unsigned.

Code	Column	Output Edit Length	Printed Results
)	TRAILING	To allow the printing of both right and left parentheses, add 2 positions.	Negative field values are enclosed in parentheses. If no filling character is specified, the left parenthesis prints before the first significant digit or decimal point, whichever is encountered first. If a leading character and this character are specified, both can print. The leading character prints inside the parentheses.
C	TRAILING	To allow the printing of CR, add 2 positions.	Prints trailing characters (CR) following a negative value; blanks follow a positive value.
D	TRAILING	To allow the printing of DB, add 2 positions.	Prints trailing characters (DB) following a negative value; blanks follow a positive value.
Z	FLOATING	DO NOT ADD A POSITION. This character will not print.	Only the significant digits of the field and any specified filling and/or trailing codes print. Decimal places, if specified, and commas are overridden. However, if decimal places are specified (even though the decimal point will not print), leading zeroes are suppressed only up to the implied decimal.
Any char. not listed above	FLOATING	Add 1 position.	Floating lead character; same attributes as $.
	FILLING	DO NOT ADD POSITIONS.	Replaces leading zeroes.
	TRAILING	Add 1 position.	Prints a trailing character following negative values.

Type C Field

Enter a number from 1 to 99 equal to the maximum value (number of characters) that can exist in the field. If the maximum value exceeds 99, do not make an entry in these columns. Decimal points, commas, and output edit codes are not used.

Character strings from a user's file are moved, truncated from the right, or padded with trailing blanks as appropriate. No other changes are made to the contents. This is true of all data that originates as character data.

When data is converted to a character string from a Z, P, or F type field by means of a PR operation, leading zeroes are suppressed. If there are decimal places in the source (Z, P, or F) field, a decimal point appears in the target (C) field. All digits following the decimal point always appear.

A zero numeric value prints as a right-justified zero if there are no decimal places in the source field or as a decimal point followed by zeroes if there are decimal places in the source field. A negative sign, if applicable, appears to the left of the first non-blank character.

When character strings are used as either transaction maximum or minimum values, only the first 15 characters are involved. When they are used for start search and end search on the RC statement, only the first 16 characters are involved. Long transaction fields or master file keys that are character strings are truncated to 15 or 16 characters.

Default edit length

VISION:Builder computes a default edit length for type C fields equal to the field length or the column heading length, whichever is longer.

Type P, Z, and F Fields

Enter a number from 1 to 30 equal to the maximum number of positions necessary to print this field. The number entered is the sum of the following items:

- Maximum value (number of digits) that can exist in the field. If decimal places are specified for the field, this count must at least equal the number of decimal places specified.
- Decimal point if decimal places are specified.
- Commas to the left of the decimal point or to the left of the low order digit when decimal places are not specified.

  The number of commas required to print a field is the integer value of the expression:

$$\frac{\text{Number of Digits} - \text{Number of Decimal Places} - 1}{3}$$

- Specified edit codes, as detailed in the table on page 4-11.

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

Default edit length

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

- The specified field length (converted to digit positions according to field type).
- For example, a type F field with specified field length of 4 bytes, when converted to digit positions, would equal 10.
- Decimal point, commas, and special edit codes.

Type E Field

Enter a number between 7 and 30 to include the two signs, decimal point, E, and two exponent digits. Floating point number (type E) fields always print in standard notation, where Xs represent the fraction and Ys the exponent:

```
+.XXXXXXX E + YY
```

If the length exceeds 13, leading blanks will print.

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

Default edit length

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

Note: The output width column on the glossaries specifies the width that VISION:Builder allows on an output report.

Processing Transaction Definitions

This technical note is designed to aid the user in processing transaction definitions.❹

Create a New Transaction Definition

The TD statement is used to define both transaction groups and transaction identifiers. Because it is allowable to have many transaction identifiers within a transaction group, many TD statements can be entered for one group. However, the following group characteristics need only be specified on the first TD statement for a group.

If the group characteristics are specified on succeeding TD statements in the group, the group characteristics are picked up from the last TD statement containing them.

VISION:Builder does not check for inconsistencies among the various TD statements in a group.

The following specifications are transaction group characteristics:

- Group name.
- Glossary specification.
- Record format specification.
- Records per block specification.
- VISION:Builder statement type TD.
- Master file name.
- Record size specification.
- Buffer size specification.

The following fields are identifier parameters and must be specified for each identifier in the group:

- Group name.
- VISION:Builder statement type TD.
- Transaction identifier 1 code.
- Identifier 1 start location.
- Identifier 1 length.
- Transaction identifier 2 code (if applicable).
- Identifier 2 start location (if applicable).
- Identifier 2 length (if applicable).
- Default create record specification (if applicable).
- Default insert segment occurrence specification (if applicable).

Note: A transaction identifier must be defined with a TD statement before field definitions (TL) for that identifier are input.

VISION:Builder completes the processing of a group definition whenever a change in the group name field occurs in the input stream. Each statement of the group definition (includes all identifiers of the group) is checked for errors and printed along with any system messages. After all statements and related messages are printed, a final message informs the user whether or not the group definition has been cataloged. The glossary listing concludes the processing of the group definition.

Delete a Field from an Existing Identifier

A field can be deleted from an existing transaction identifier definition by entering a TD statement containing:

- Existing group name.
- VISION:Builder statement type TD.
- Existing master file name.

The TD statement is followed by a TL statement containing:

- Existing group name.
- Transaction identifier 1.
- Delete specification.
- Field length.
- VISION:Builder statement type TL.
- Transaction identifier 2 (if applicable).
- Field location.
- Field name.

Change Characteristics of an Existing Field

A change to any of the existing field characteristics can only be accomplished by deleting the existing field and adding the redefined field to the existing transaction identifier definition. (See Delete a Field from an Existing Identifier on page 4-16 and Add a Field to an Existing Transaction Identifier on page 4-19.)

Delete an Existing Group Definition

An existing transaction group definition (including all identifiers for that group) can be deleted by entering a TD statement containing the following information:

- Existing group name.
- VISION:Builder statement type TD.
- Delete specification.
- Existing master file name.

Delete an Existing Identifier from the Group Definition

An existing entire transaction identifier definition can be deleted by entering a TD statement that contains the following information:

- Existing group name.
- Transaction identifier code 1.
- Delete specification.
- VISION:Builder statement type TD.
- Transaction identifier code 2 (if applicable).
- Existing master file name.

Update an Existing Transaction Definition

Updates or changes can be made to an existing transaction definition that do not necessitate a complete redefinition of the transaction group or transaction identifier.

- The TD statement is used to define both transaction groups and transaction identifiers.
- Because it is allowable to have many transaction identifiers within a transaction group, many TD statements can be entered for one group. VISION:Builder completes the processing of a transaction definition whenever a change in the transaction group name field occurs.
- Detailed in the following sections are typical areas of change. Each section describes the input necessary to obtain the desired results. At times it might be necessary to combine several of these changes to obtain the desired results.
- Updates or changes can be made during any definition/maintenance run.

Add a New Transaction Identifier to an Existing Group

A new transaction identifier can be added to the existing transaction group by entering a TD statement containing the following information:

- Existing group name.
- VISION:Builder statement type TD.
- Transaction identifier 1 code.
- Identifier 1 start location.
- Identifier 1 length.
- Transaction identifier 2 code (if applicable).
- Identifier 2 start location (if applicable).
- Identifier 2 length (if applicable).
- Default create record specification (if applicable).
- Default insert segment occurrence specification (if applicable).

The TD statement is followed by TL statement(s) defining the fields within the identifier.

Change the Record Format Characteristics of an Existing Group

The record format characteristics are the aggregate of record format, record size, records per block, and buffer size. If any of these fields are to be changed, enter the following information:

- Existing group name.
- Existing glossary specification.
- Existing or new record format.
- Existing or new records per block (if applicable).
- VISION:Builder statement type TD.
- Existing master file name.
- Existing or new record size (if applicable).
- Existing or new buffer size (if applicable).

Change the Glossary Level for a Transaction Group

The glossary can be temporarily changed to be printed in a new format. This change is in effect only until the end of the definition maintenance run, at which time the glossary level reverts back to the original specification. To temporarily change the glossary level, enter a TD statement containing the following information:

- Existing group name.
- VISION:Builder statement type TD.
- New temporary glossary specification.
- Existing master file name.

The glossary can be permanently changed from the existing specification by redefining the record format characteristics. Enter a TD statement containing the following information:

- Existing group name.
- New glossary specification.
- Existing record format.
- Existing records per block (if applicable).
- VISION:Builder statement type TD.
- Existing master file name.
- Existing record size (if applicable).
- Existing buffer size (if applicable).

Change Default Specification of an Existing Transaction Identifier

The above specifications can be added to or changed by deleting the existing transaction identifier, redefining the identifier to include these specifications, and reentering all TL statements for the identifiers being processed. (See Delete an Existing Identifier from the Group Definition on page 4-16 and Add a New Transaction Identifier to an Existing Group on page 4-17.)

Change Start Location and/or Length of an Existing Transaction Identifier

A change to either or both of the above specifications can be accomplished by deleting the existing transaction identifier, redefining the identifier to include these specifications, and reentering all TL statements for the identifier being processed. (See Delete an Existing Identifier from the Group Definition on page 4-16 and Add a New Transaction Identifier to an Existing Group on page 4-17.)

Add a Field to an Existing Transaction Identifier

A field can be added to the existing transaction identifier definition by entering a TD statement containing the following information:

- Existing group name.
- VISION:Builder statement type TD.
- Existing master file name.

The TD statement is followed by a TL statement containing:

- Existing group name.
- Transaction identifier 1.
- Field location.
- Field type.
- Action code.
- Minimum value (if applicable).
- VISION:Builder statement type TL.
- Transaction identifier 2 (if applicable).
- Field length.
- Field name.
- Decimal places (if applicable).
- Maximum value (if applicable).

Catalog Operations

This technical note contains additional information on the SAV, DEL, and INS operations for the library maintenance (CT) statement.

SAV Operator

- If the SAV operator is used with no operands, all requests in the input stream and all requests retrieved with CR statements become members of the group.
- If any request written (explicitly or implicitly) as an operand of a SAV statement is found to be invalid, the request group is not cataloged. However, all such valid requests are cataloged.

DEL Operator

The three types of DEL operations are:

Type 1 — Delete entire group (that is, positions [1-8] have a group name and positions [14-61] are blank).

If a type 1 delete is requested, the named group is deleted, along with the references to the given elements in the given group. The elements themselves are not deleted from the library; they must be deleted separately.

Type 2 — Delete one or more elements of a given group (that is, positions [1-8] have a group name and positions [14-61] have at least one entry).

If a type 2 delete is requested, the references to the given elements in the given group will be erased, but the elements themselves are not deleted from the library.

Example:

GROUP A contains:
REQUEST 1
REQUEST 2
GROUP B
REQUEST 3
GROUP C

If a DEL operator has GROUP A in positions [1-8] and REQUEST 1 GROUP C in positions [14-29], the results will be as follows:

Example:

GROUP A contains:
REQUEST 2
GROUP B
REQUEST 3

However, REQUEST 1 and GROUP C are not deleted from the library.

Type 3 — Delete one or more elements from each group in which they appear (that is, positions [1-8] are blank and positions [14-61] have at least one entry).

If a type 3 delete is requested, all references to the named operand are erased in each group in which they appear.

If the operand is a request, the request is deleted from the library as well.

If the operand is a group, the group is not deleted from the library.

In summary, to purge a request from the library, a type 3 DEL must be used. To purge a group from the library, a type 1 DEL and a type 3 DEL must be used.

INS Operator

Only entries in operands 1 and 2 are considered; entries in operands 3 and 4 are ignored.

Sequence of CT Statement Operations

The following is the sequence in which CT statement operations are performed:

1. Request dumps performed (DMP).
2. Requests that are to be replaced or added to the library are cataloged (unless request is invalid).
3. Group names and element names of SAV and INS operators that do not reference requests or groups in the input stream are resolved using the groups and requests that have been previously cataloged.
4. All DEL operations and then all INS operations are performed against the group saved (in this run).
5. All DEL operations and then all INS operations are performed against the groups saved (from previous runs).
6. The LST operation is performed.

Note: If there is an error in the continuation statement, the next statement is ignored completely if the operation is blank on that statement.

Chapter

5

Graphic Fixed Format Statement Listing

This chapter contains a description of each VISION:Builder statement. The VISION:Builder statements are as follows:

En Statement - Graphics

The graphics output format statement (En statement) specifies the format of the graph, the graph data report, and the statistics report. The En statement also specifies whether subfiles are output and their format.

Position	Entry
01-08	Request name
09-10	Statement type
13	Print 8 lines per inch?
14-16	Width of page
17-19	Height of page
21	Special forms
22-23	Maximum number of lines per page
24-27	Maximum number of pages
28	Page title at bottom of page?
29	Column heading type
30	Column heading position
31-32	Date position
33-34	Page number position
35-38	Start page numbers at
39	Line numbers
40	Labels on summary lines
41-56	File entries
41-42	Report handling
43	Entire record selection
44-51	DTF/ddname or file name
52-55	Subfile blocking factor
56	Subfile format
57	Column heading character
58	Empty field control
59	Autogrand?
62	Invalid summary indicator
72	Continue

Request name (1-8)

Required entry

Identifies this statement and others accompanying it as one request. See general rule A. Enter the same request name for all statements in the graph group.

Statement type (9-10)

Required entry

Identifies an output format statement and connects it to a graph report group.

Enter an E followed by a number (n) from 1 to 9 to relate this statement to a specific report or subfile. The number must be the same on all En, Kn, Rn, Pn, and Tn statements in a graph report statement group.

Note: Each request can accommodate up to 9 reports, each in a different format.

Print 8 lines per inch? (13)

Computes the page size to print 8 lines per inch instead of 6.

Entry	Result
Blank or N	6 lines/inch.
Y	8 lines/inch.

Width of page (14-16)

Specifies the width of the printed page. Enter, right-aligned, a number to define the logical page width (in characters). The minimum page width for a graphics report is 120 characters.

Entry	Page Width
Blank	M4PARAMS default page width.
Number	Enter a number from 1 to installation standard as set by M4PARAMS.

Note: Fields extending beyond the specified width fold or print on the next line of the data report.

Height of page (17-19)

Specifies the logical height of the printed page. The page height must accommodate margins, titles and the graph control field (subtitle) with intervening blank lines, and the graph and data or statistics report.

		Maximum Number of Lines	
Code	Height In Inches	6 Lines/Inch	8 Lines/Inch
Blank	Installation Standard	Installation Standard	Installation Standard
A	8.50	51	68
B	11.00	66	88
C	3.25	19	26
E	22.00	132	176

Note: Code C: For installations without floating point option, the minimum page height is 12 lines for scatter diagrams; with floating point option, it is 22 lines. For bar graphs, the minimum page height is 12 lines.

Specify any other page height from the minimum to 255 (to represent the number of single spaced lines per page).

Special forms? (21)

Request forms other than installation standard paper. This specification has no effect on systems using spooled output.

Entry	Result
Blank or N	Installation standard forms.
S or Y	Special forms.

Note: When S or Y is specified, a message prints to alert the operator to mount special forms.

Maximum number of lines per page (22-23)

Limits the number of lines to print per page.

Entry	Result
Blank	Maximum number of lines possible depending on page height.
Minimum number possible to 99	Maximum number of lines to be printed on each page of the graph, data, or statistics report. Cannot exceed Height of Page minus one.
NM	No margin will be subtracted from the page height for the report, and the full page height will be used.

Note: Code C: For installations without floating point option, the minimum page height is 12 lines for scatter diagrams; with floating point option, it is 22 lines. For bar graphs, the minimum page height is 12 lines.

Maximum number of pages (24-27)

Limits the total number of printed pages for a report but does not limit the data selected for output.

Entry	Result
Blank	All selected data is printed for the graph, data, and statistics report.
1 to 9999	Limits the number of pages to the value specified.

- Data that would normally be printed after the number of pages indicated is not printed.
- Graphics output occurs in the following sequence:

1. The data report (if not suppressed).
2. The graph(s).
3. The statistics report.

If the maximum number of pages is reached while generating the data report, neither the graph(s) nor the statistics report is printed.

Page title at bottom of page? (28)

Specifies the position for printing the report title specified on the Tn statement.

Entry	Title Position
Blank or N	Top of page.
Y	Bottom of page.

Column heading type (29)

Specifies the column headings for the data report or the captions on the graph axes.

Entry	Result
Blank	As shown on file definition or temporary field definition.
F	Field names (Rn statement) become column headings and captions.
X	No column headings or captions are printed on the report or graph.

Column heading position (30)

Specifies the placement of the column headings on the data report.

Entry	Column Heading Position (Centered or Under Columns of Data)
Blank or T	Top of page.
B	Below detail items and summaries.

Date position (31-32)

Specifies date placement on all graphics output.

Entry	Result
Blank or UL	Upper left corner of page.
UR	Upper right corner of page.
LL	Lower left corner of page.
LR	Lower right corner of page.
MT	Middle top of page.
MB	Middle bottom of page.
ND	No date.

Note: If date position is specified, the report date must also be specified on the ER statement [11-16] or the date does not print.

Page number position (33-34)

Specifies page number placement on all graphics output.

Entry	Page Number Position
Blank or UR	Upper right corner of page.
UL	Upper left corner of page.
LL	Lower left corner of page.
LR	Lower right corner of page.
MT	Middle top of page.
MB	Middle bottom of page.
NP	No page numbers.

Note: If both page number and date are specified for the same page location, only the page number prints.

Start page number at (35-38)

Specifies the page number for the first page of the graphics output. Page numbering starts with the data report (if not suppressed). The maximum starting number allowed is 999,999.

Entry	Result
Blank or 1	Page numbering begins at 1.
2 to 9999	Page numbering begins at the number specified (useful for special or partial reports).
PAGE	Resets the page number to 1 each time a graph control field changes value.

Note: The preface page (Pn statement) is not numbered.

Line numbers (39)

Specifies and positions line numbers on a report. Line numbers begin at and run consecutively to 999,999. Lines are counted and numbered by logical detail lines rather than actual printed lines. Therefore, if the end line specification is used (Rn statement [25]) or if the line is folded, it is counted as only one line.

Entry	Result
Blank or N	No line numbers.
L	Print line number at left side of page beginning at 1.
R or Y	Print line numbers at right side of page beginning at 1.
B	Print line numbers at both sides of page beginning at 1.

- If line numbers are specified, six positions are reserved for them. If you also specify width of page [14-16], you must specify a page width large enough to accommodate both the detail line and the line number.
- On folded lines, left line numbers print on the left of the first line, right line numbers on the right of the last of the folded lines.
- Line numbers are reset whenever page numbers are reset.

Labels on summary lines (40)

Valid for bar graph data reports only.

Positions or suppresses the 18 spaces of the summary descriptions (labels) when summaries are requested (Rn statement [31-33]).

Entry	Result
Blank	The 18 spaces are reserved at the left for the summary labels.
L	Columns of data begin at the left of the detail report.
X	No summary labels. If summaries are requested, they print left-aligned without descriptions.

- Of the 18 spaces, [1-8] are reserved for the field summarized, [10-14] are for the type of summary, and [16-17] for the category ID.
- If a summary is requested on columns in spaces [1-18], the label is overlaid by the summary. Therefore, position your data so that those with summaries begin no earlier than position [19].

Report handling (41-42)

Specifies whether alternate report files, subfiles, alternate M4LIST files, and file definitions are generated.

Entry	Result
Blank	Standard or graphics report and subfiles, if requested, are generated.
RF	Report is placed on a separate report file.
DR	Subfile, subfile file definition, and report are created.
AL	This report and the associated report START, END, NO DATA, NUMBER OF LINES OF PRINT, NUMBER OF MISSING/INVALID/OVERFLOW FIELDS, and INCOMPLETE SUMMARY messages are printed on the alternate M4LIST. All signons, source statements, messages, and statistics are printed on M4LIST.

- File definition statement images are written on the subfile specified on the RF statement having a C in position 53.
- Entries of NR or DN are not allowed.
- An entry must be made in the DTF/ddname or file name specification [44-51]. The same name must be entered as an additional file on the RF statement defining that name as applying to one of the alternate report files or subfiles.

Entire record selection (43)

Identifies special handling of selected input file records or data for subfiles.

Entry	Output of Entire Selected Records to Subfiles
Blank	The fields specified on the Rn statement produce a report or subfile.
O or 0	Output selected old master file records to a subfile.
N[4]	Output selected new master file records to a subfile.
1 to 9	Output selected coordinated file records (1 to 9) or array records to a subfile.
X	Output selected transaction file records to a subfile during transaction request processing.[4]
A, B, E, G, H, J, K, M, Q	Output selected array records to a subfile. Code must match the one listed on the RF statement.

Note: Blank is not valid if generating IMS subfiles.

See En Statement - Output Specification on page 2-40 for more information.

DTF/ddname or file name (44-51)

Required entry

Identifies the subfile, alternate M4LIST file, or alternate report file produced from this request. See general rule A. (This name must also be entered on an RF statement if it is a subfile or alternate report file.)

Non-DL/I files and alternate report files

Enter a unique 8-character name (must match RF).

DL/I files (subfiles only)[I]

Enter a user defined DBD name (up to 8 characters and must match RF).

- If a file definition is created, it is identified by this name.
- The same subfile name can be used by more than one request, allowing output to the same subfile for various selection criteria. Whenever a subfile is produced from more than one request, all requests must have the same entries in the subfile blocking factor and subfile format specifications.

Alternate M4LIST

Entry	Result
Blank	M4LIST1 is used as the alternate M4LIST DTF/ddname for the report.
Unique name up to 8 characters (7 characters for VSE)	This unique name is used as the alternate M4LIST DTF/ddname for this report. It cannot appear on an RF statement but must be defined in the JCL. For VSE, SYS055 will be used as the SYS number for all alternate M4LISTs unless they are overridden by RP statements.

Subfile blocking factor (52-55)

Describes the block size of physical records on the subfile as follows. (If multiple requests refer to the same subfile, the attributes must be identical.)

The following information on the values that can be specified for the Subfile blocking factor [52-55] entry are based on the value that is specified in the Subfile format [56] entry.

Variable blocked (V or blank in subfile format (56))

This value represents the buffer size and must be large enough to accommodate the largest record output. If the operating system supports system-determined block size and the file meets the criteria for this capability, this value is ignored and is determined by the operating system.

Entry	Entire Record Selection Entry?	Result
Blank	Yes	Value defaults to that specified in the input file.
Blank	No	Value defaults to installation-specified block size for M4REPO (see M4PARAMS in the *VISION:Builder for OS/390 Installation Guide* or the *VISION:Builder for VSE Installation Guide*).
1 to 9999		Buffer size.
1H to 327H		Buffer size, interpreted as 100 to 32,700.
nnnK		Multiples (nnn) of 1024 bytes. If the value is greater than 31K, interpreted as variable blocked spanned record size.

Fixed blocked (F in subfile format (56))

Entry	Result
Blank	Fixed blocked, if the operating system supports system-determined block size and the file meets the criteria for this capability. Fixed unblocked, otherwise.
1 to 9999	Number of records per block.
1H to 327H	Number of records per block, interpreted as 100 to 32,700.
nnnK	Multiples (nnn) of 1024 bytes (1K to 31K).

- VISION:Builder assumes a value of 1 if entire record selection is specified.
- To create a fixed unblocked subfile under an operating system that supports system-determined block size, enter a 1 for subfile blocking factor or code RECFM=F on the JCL statement for the subfile.

Undefined (U in subfile format (56))

Entry	Entire Record Selection Entry?	Result
Blank	Yes	Value defaults to that specified for the input file.
Blank	No	Value defaults to 8 less than the installation-specified block size for M4REPO.
1 to 9999		Buffer size.
1H to 327H		Buffer size, interpreted as 100 to 32,700.
nnnK		Multiples (nnn) of 1024 bytes (1K to 31K).

ISAM fixed blocked (I in subfile format (56))

Entry	Result
Blank	VISION:Builder assumes value of 1 if entire record selection is specified.
1 to 9999	Number of records per block.
1H to 327H	Number of records per block, interpreted as 100 to 32,700.
nnnK	Multiples (nnn) of 1024 bytes (1K to 31K).

Note: Unblocked ISAM files are not processed by VISION:Builder.

ISAM variable blocked (J in subfile format (56))

Not valid for VSE.

Entry	Entire Record Selection Specified?	Result
Blank	Yes	The buffer size of the entire record selection file is used.
Blank	No	Report file buffer size is used.
1 to 9999		Buffer size.
1H to 327H		Buffer size, interpreted as 100 to 32,700.
nnnK		Multiples (nnn) of 1024 bytes (1K to 31K).

Packed (P in subfile format (56))

Entry	Entire Record Selection Specified?	Result
Blank	Yes	The buffer size of the entire record selection file is used.
Blank	No	Report file buffer size is used.
1 to 9999		Buffer size.
1H to 327H		Buffer size, interpreted as 100 to 32,700.
nnnK		Multiples (nnn) of 1024 bytes. If the value is greater than 31K, interpreted as variable blocked spanned record size.

VSAM (E or K in subfile format (56))

Leave blank.

Relational table generation (S in subfile format (56))

Leave blank.

IMS (D or H in subfile format (56))

Leave blank.

Subfile format (56)

Describe the format of the subfile's physical records. If multiple requests refer to the same subfile, the attributes must be identical.

Entry	Record Type
Blank or V	Variable blocked.
F	Fixed blocked.
U	Undefined.
I	ISAM fixed blocked.
J	Variable ISAM (not supported by all operating systems).
P	Packed (variable blocked).
D	IMS subfile.
K	Key sequenced VSAM.
E	Entry sequenced VSAM.

- You can write to the same subfile from different requests. The subfile name is the common identifier.
- If subfiles are specified with the same name but different format or blocking parameters, the subsequent requests are invalid and no processing takes place.
- When using F format subfiles, data from different requests for the same subfile must have the same length (that is, for each request, the total length of the fields in the Rn statement must be the same).

Column heading character (57)

Overrides the installation standard column heading borders on the data and statistics reports (see M4PARAMS in the *Installation Guide*).

Entry	Border
Blank	Installation standard.
B	Blank lines replace regular borders.
N	No borders (blank lines will not be output in place of borders).
Any other character	The character specified is used for column heading borders instead of the installation standard.

Empty field control (58)

Specifies the results of "empty" fields in summaries for bar graphs only. (An "empty" character string field contains blanks; an "empty" numeric field contains zeroes.) Enter one of the following, depending on how you want your summaries computed:

Entry	Blank Character String Fields Included in Summaries and Counts of Data Values in the Statistics Reports?	Zero Numeric Fields Included in Summaries and Counts of Data Values in the Statistics Reports?
Blank	No	Yes
I	Yes	Yes
E	No	No

Autogrand? (59)

Specifies whether an automatic grand summary page is to be printed for a specific bar graph data report.

Entry	Result
Blank	No change from M4PARAMS (see the *VISION:Builder for OS/390 Installation Guide* or the *VISION:Builder for VSE Installation Guide*)) or from the selection on the RP statement.
N	No autogrand page is printed.
Y	Autogrand page is printed.

Invalid summary indicator (62)

Indicates whether or not detail report summaries are to be flagged if they contain incomplete results.

Entry	Result
Blank	No flagging of summaries.
Any other Character	Summary will be flagged with this character.

Note: A plus sign (+) prints if there is not enough room to print the complete summary value.

Continue (72)

Indicates multiple En statements. Continuation is used for additional file entries [41-56] only.

Enter any non-blank character to specify continuation of the En statement.

Jn Statement - Graphics

The graph specification (Jn) statement defines the output of the request as one with graphics report output. It specifies the type of graph produced and the plot and axis characters used.

Position	Entry
01-08	Request name
09-10	Statement type
11-12	Graph type
13	Suppress data report?
14	Bar type
15-16	Bar width
17	Primary plot character
18	Secondary plot character
19	Fit plot character
20	Horizontal axis character
21	Horizontal hash character
22	Vertical axis character
23	Vertical hash character

Request name (1-8)

Required entry

Identifies this statement as part of a particular request. See general rule A. Enter the same name for all statements in the request.

Statement type (9-10)

Required entry

Identifies the statement as a graph specification and connects it to a particular requested graph output.

Enter J and the applicable number (n) from 1 to 9. The number must be the same on all En, Kn, Rn, Pn, and Tn statements in a graph report statement group.

Graph type (11-12)

Required entry

Specifies the type of graph produced. Enter one of the following graph types:

Graph	Type Description
HD	Horizontal bar graph with discrete categories.
HR	Horizontal bar graph with ranges for categories.
VD	Vertical bar graph with discrete categories.
VR	Vertical bar graph with ranges for categories.
SN	Scatter diagram with no fit plotted.
SL	Scatter diagram with linear fit plotted.
SS	Scatter diagram with second order fit plotted.

Suppress data report? (13)

Specifies whether a graph data report will be generated.

Entry	Result
Blank or N	Graph data report is printed.
Y	Graph data report is suppressed.

Bar type (14)

Bar graph only

Specifies the type of bar graph generated. This entry is valid only when a bar graph type is specified (HD, HR, VD, or VR).

Entry	Result
Blank or B	Both absolute and relative bar graphs are generated.
A	Absolute bar graph is generated.
R	Relative bar graph is generated.

Bar width (15-16)

Bar graph only

Specifies the number of characters printed for the width of each bar. This entry is valid only when a bar graph is specified (HD, HR, VD, or VR).

Entry	Result
Blank	A default bar width of 1 is assigned.
1 to 99	The number represents the number of characters that make up the width of each bar.

- When the specified bar width is too large to allow all bars to be printed on the graph, VISION:Builder calculates the maximum bar width that will permit all bars to be graphed and uses that.
- If a width of 1 will not permit all bars to be graphed, the number of bars printed on the graph is reduced.

Primary plot character (17)

Specifies the primary character used in plotting values on a graph.

Bar graphs	This character plots every other bar, beginning with the top bar on horizontal bar graphs and the leftmost bar on vertical bar graphs.
Scatter diagrams	This character plots single points on a graph representing a single X, Y pair plotted at that location. Enter any character or leave blank. If left blank, the default character is used on the graph.

Note: An X is the default as delivered, specified in M4PARAMS.

Secondary plot character (18)

Specifies the secondary plot character used in plotting values on a graph.

Bar graphs	This character plots every other bar, alternating with the primary plot character bars. If left blank, the character specified as the primary plot character is used for all bar graphs.
Scatter diagrams	This character plots points representing more than one X, Y pair plotted at the same location. Enter any character or leave blank. If left blank, the default character is used as the secondary plot character on a scatter diagram.

Note: An asterisk (*) is the default as delivered, specified in M4PARAMS.

Fit plot character (19)

Scatter diagram only

Specifies the character used in plotting the points for the linear or second order fit equation on a scatter diagram. This entry is valid only when a scatter diagram with a linear fit (graph type SL) or a second order fit (graph type SS) is specified.

Enter any character or leave blank. If left blank, the system default character is used as the plot character.

Note: A period (.) is the default as delivered, specified in M4PARAMS.

Horizontal axis character (20)

Specifies the character used in forming the horizontal axis on a bar graph or scatter diagram. Enter any character or leave blank. If left blank, the default character is used to form the horizontal axis.

Note: An underscore (_) is the default as delivered, specified in M4PARAMS.

Horizontal hash character (21)

Scatter diagram only

Specifies the character (hash mark) that delineates the intervals on the horizontal axis (X axis) of a scatter diagram. Enter any character or leave blank. If left blank, the default character is used to designate the intervals on the horizontal axis of a scatter diagram.

Note: A vertical bar (|) is the default as delivered, specified in M4PARAMS.

Vertical axis character (22)

Specifies the character used in forming the vertical axis on a bar graph or scatter diagram. Enter any character or leave blank. If left blank, the default character is used as the vertical axis.

Note: A vertical bar (|) is the default as delivered, specified in M4PARAMS.

Vertical hash character (23)

Scatter diagram only

Specifies the character (hash mark) that delineates the intervals on the vertical axis (Y axis) of a scatter diagram. Enter any character or leave blank. If left blank, the default character is used to designate the intervals on the vertical axis of a scatter diagram.

Note: A dash (-) is the default as delivered, specified in M4PARAMS.

Kn Statement - Graphics

The axis specification statement (Kn statement) defines the characteristics of an axis on a graph. The Kn statement is used for the X and Y axes of a scatter diagram and for the category axis of a range (types HR and VR) bar graph. It is invalid for discrete bar graphs (types HD and VD).

Position	Entry
01-08	Request name
09-10	Statement type
11	Graph modifier
12-59	Interval control
12-27	START interval
28-43	END interval
44-59	DELTA (increment)

Request name (1-8)

Required entry

Specifies the request to which this graph statement belongs. See general rule A. Use the same request name for all statements in the request.

Statement type (9-10)

Required entry

Identifies an axis specification and connects it to a particular graph statement group. Enter K and a number (n) from 1 to 9. The number must be the same on all En, Jn, Rn, Pn, and Tn statements in a graph report statement group.

Graph modifier (11)

Required entry

Identifies the axis being defined with this Kn statement. Both the X and Y axes must be defined for a scatter diagram. For range bar graphs (graph type HR and VR), only the category axis is defined. Enter one of the following graph modifiers:

Modifier	Description
C	Category axis. Defines the category axis for a range bar graph (graph type VR or HR).
X	X axis. Defines the X axis for a scatter diagram.
Y	Y axis. Defines the Y axis for a scatter diagram.

Interval control (12-59)

Specifies the range and intervals into which the axis is divided. START, END, and DELTA entries are required for defining X and Y axes of scatter diagrams and the category axis of range bar graphs.

- The values on an axis extend from (and include) the START value to (and exclude) the END value. The number of increments within that range is specified by the DELTA value.
- START, END, and DELTA must be specified so that the difference between START and END, divided by the DELTA value, equals a whole number.
- START and END restrict the range of data that can appear on the graph. DELTA serves as a device for label positioning.

START interval (12-27)

Required entry

Specifies the starting value of the axis being defined. Any number equal to START is considered in range. This entry is an inclusive end point and is the first label on the axis specified by the graph modifier.

Enter any numeric value, signed or unsigned, that is the starting value of the X or Y axis (scatter diagrams) or of the ranges on the category axis (range bar graph). The START value can be less or greater than the END value. Decimal points and commas are allowed.

When the START value is less than the END value, the numbers or ranges along the axis and the values on the data report are ascending. When the START value is greater than the END value, the numbers or ranges along the axis and the values on the data report are descending.

Note: START and END values cannot be equal.

END interval (28-43)

Required entry

Specifies the end value of the axis defined. Any number equal to END is considered out of range and out of range values cannot be graphed. This entry is an exclusive end point.

Enter any numeric value, signed or unsigned, that indicates the end value of the X or Y axis or of the ranges on the category axis. The END value can be greater or less than the START value. Decimal points and commas are allowed.

When the END value is greater than the START value, the numbers or ranges along the axis, and the values on the data report are ascending. When the END value is less than the START value, the numbers or ranges along the axis, and the values on the data report are descending.

Note: END and START values cannot be equal.

DELTA (increment) (44-59)

Required entry

Specifies the size of one interval on the X or Y axis of one range on the category axis. The difference between the START and END values is divided by the DELTA value to determine the number of intervals or categories on the axis. The resulting number of intervals must be a whole number.

Enter any numeric value other than zero. The value can be signed, but the sign will be ignored. Decimal places and commas can be used.

Rn Statement - Graphics

The Rn statement is used to specify the data fields for the graphics output or subfiles (graph data report, the graphs, and statistics report). One graph report statement group can contain a minimum of two and a maximum of three Rn statements. Bar graphs require category and summary axis Rn statements. Scatter diagrams require X and Y axis statements.

Note: If an incorrect number of Rn statements is included in the run, the request is invalid.

Position	Entry
01-08	Request name
09-10	Statement type
11-13	Sequence number
14-15	Number of spaces before column
16	Qualifier
17-24	Field name
25	End line?
26	Non-print
31-33	Summary entries
31	Total
32	Cumulative
33	Count
37	Graph modifier
53-67	Output edit
68-71	Partial field
68-69	Starting character
70-71	Numer of characters

Request name (1-8)

Required entry

Identifies this statement and others accompanying it (if any) as one request. See general rule A. Enter the same request name for all statements in the request.

Statement type (9-10)

Required entry

Identifies a reporting statement. Enter an R followed by a number (n) from 1 to 9. The number must be the same on all En, Jn, Kn, Rn, Pn, and Tn statements in a graph report statement group.

Sequence number (11-13)

Specifies the sequence of the statements for the data report and determines left to right sequence of the data on the report or subfile. See general rule I. If sequence numbers are left blank, the order of input is assumed. Assign up to 3 alpha, numeric, or alphanumeric characters in the order in which the data is to be printed across the report or output to a subfile. Sequencing is in ascending order.

Number of spaces before column (14-15)

Specifies the number of blank spaces to the left of the value of the field on the report.

Entry	Result
Blank	2 spaces before the field (VISION:Builder default).
0 to 99	0 to 99 spaces before the field.

- X, Y, C, S graph modifiers - specification affects printing of fields on the data report only.

  G graph modifier - specification affects all graphic report outputs.

- This specification does not apply to the subfiles created.

Qualifier (16)

Identifies the type of field or file where the field exists.

Code	Location of File or Field
Blank or N	New master file field.
O or 0	Old master file field.
1 to 9	Coordinated files 1 to 9 field.
T	Temporary field previously defined.
F	Flag field.
X	Transaction file field.❹
W	Working storage.
V	Linkage section.
A, B, E, G, H, J, K, M, Q, 1 to 9	Array field (must match qualifier listed on the RF statement).

Field name (17-24)

Required entry

Identifies the field output on a report or written to a subfile. See general rule A. When combined with a graph modifier [37], the field name identifies fields to serve various functions in a graph report.

- *Graph control statement* (G in [37]) - The field name identifies this field as a graph control field. A data report, graph, and statistics report are produced for each different value of this field. This field name is printed as a page subtitle on all graphics output.

  Enter any field name on which control breaks could be taken.

- *Bar graph category axis statement* (C in [37]) - The field name identifies the value defining the categories along the category axis.

  Range bar graphs - Specify a numeric field (types Z, P, or F).

  Discrete bar graphs - Any numeric or alphanumeric field can be specified (types Z, P, F, or C).

- *Bar graph summary axis statement* (S in [37]) - The field name identifies the value defining the field on which to take the specified type of summary (types Z, P, F, or C).

- *Scatter diagram x axis statement* (X in [37]) - The field name identifies the value to plot along the X axis (types Z, P, or F).

- *Scatter diagram y axis statement* (Y in [37]) - The field name identifies the value to plot along the Y axis (types Z, P, or F).

End line? (25)

Prints data from each report record on more than one line of the data report.

Entry	Result
Blank or N	Contents of next field print on the same line.
Y	Contents of next field print on the following line.

Note: If the end line is specified on any field(s), those column heading(s) also fold to a new line.

Non-print (26)

Suppresses printing of field values in a report.

Entry	Result
Blank or N	Contents of the field is printed.
Y	Printing of contents of field is suppressed.

- For a Y entry: X, Y, and C graph modifiers suppress printing of fields on the data report and captions on the graph.
- For a Y entry: G graph modifiers suppress printing of fields for all graphics output.
- For a Y entry: S graph modifiers omit detail data on the data report but print summaries.

Summary entries (31-33)

The summary entries identify the type of summary taken on the specified field for each category on a bar graph.

If summaries are requested, the graph modifier [37] must be S. Only one summary is allowed per bar graph.

Total (31)

Entry	Result
X	A total of the valid values of the field is taken for each category.

Cumulative (32)

Entry	Result
X	Designates cumulative totals for the field from the beginning of the report for each category.

Count (33)

Entry	Result
X	Designates a count of the field for each category.

Graph modifier (37)

Required entry

Enter a graph modifier based on the following table.

Graph Mod.	Description	Graph Type	Statement	Field Type	Function
G	Graph Control	All Types	Rn	Any field on which a control break can be specified.	Generates separate graphics output when the field value changes; prints the value as a page subtitle.
C	Category Axis	Range Bar: HR & VR	Kn		Specifies the interval controls along the category axis.
		Range Bar: HR & VR	Rn	Field types Z, P, F. Must be numeric.	Identifies the field whose values are grouped in ranges and output along the category axis.
		Discrete Bar: HD & VD	Rn	Field types Z, P, F, C. C type is treated as character.	Identifies the field whose values are output along the category axis each time the value of the field changes.
S	Summary Axis	Discrete or Range Bar: HD, VD, HR, & VR	Rn (one summary entry must contain X)	Field types Z, P, F, C.	Identifies the field whose values are summarized for the category and illustrated on the graph.
X	X Axis	Scatter Diagram: SN, SL, SS	Kn		Specifies the interval controls along the X axis.
		Scatter Diagram: SN, SL, SS	Rn	Field types Z, P, F.	Identifies the field whose values are plotted along the X axis.
Y	Y Axis	Scatter Diagram: SN, SL, SS	Kn		Specifies the interval controls along the Y axis.
		Scatter Diagram: SN, SL, SS	Rn	Field types Z, P, F.	Identifies the field whose values are plotted along the Y axis.

Output edit (53-67)

Overrides the edit specification on the L0 statement. Please refer to the output specification (Rn) statement for more details on output edit.

- For graph modifiers X, Y, C, and S, this specification overrides the edit specifications from the file definition or temporary field definition when printing fields on the data report.
- For graph modifier G, this specification affects all graphics report output.

Partial field entries (68-71)

Allows the processing of any part of a character string field (type C). This specification applies to both reports and subfiles.

- If partial fielding is used, both starting and number of characters must be specified and must be within the range of the field.
- Partial field flags cannot be used in Rn statements.
- Only valid for graph modifiers G, C, or S.

Starting character (68-69)

Identifies the leftmost position of the starting character in a partial field. Enter a number from 1 to 99 to identify this position.

Number of characters (70-71)

Indicates the number of characters to be output from a partial field. Enter a number from 1 to 99 to reflect this.

Tn Statement - Graphics

The title lines are optional for a graph. They are used in the same way as for other output.

Position	Entry
01-08	Request name
09-10	Statement type
11-13	Sequence number
14-72	Text

Request name (1-8)

Required entry

Identifies the request to which this statement belongs. See general rule A. Enter the same request name for all statements in the request.

Statement type (9-10)

Required entry

Identifies a title and connects it to other statements in the graph statement group. Enter the letter T and the same number (n) from 1 to 9. The number must be the same on all En, Kn, Rn, Pn, and Tn statements in a graph report statement group.

Sequence number (11-13)

Specifies the sequence of the lines of title on the graphics output. See general rule I. If blank lines appear in the title, sequence numbers must be assigned on all lines of title. Positions [14-72] in each line must contain either the text for a line or a delimiter in position [14] if it is a blank line.

Entry	Result
Blank	Order of statement input is assumed.
Up to 3 alpha, numeric, or alphanumeric characters	Order that title lines appear on the data and statistics report. Sequencing is in ascending order.

Note: Duplicate sequence numbers are accepted in order of input.

Text (14-72)

Specifies the title that appears on the graphics output. Enter the text required for the title.

- A line of text is scanned until a delimiter is encountered. (The scan will continue over several lines, if necessary.)
- Title lines are normally centered at the top of the page between the date and page number. This default can be overridden by an entry on the En statement.
- Only 59 characters of text are defined per statement. It may be necessary to use multiple statements to define a title with text that is longer than 59 characters.

Index

Symbols

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Z