

Advantage[®] VISION:Transact[™] for CICS[®]

Specifications Guide

Release 7.8



Computer Associates[®]

B02570-1E

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

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

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

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

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

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

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

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

© 2004 Computer Associates International, Inc.

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

Contents

Chapter 1: Introduction

Overview	1-1
Definition Run (Catalog Maintenance)	1-3
Run Control Group(RC, RP, IT)	1-4
File Definition Statements (FD, LS, LR, L0, Ln)	1-5
Screen/COMMAREA Definition Statements (SD, SL, SS, S0, Sn)	1-5
Request Processing Statements (ER, TF, PR)	1-6
Catalog Request Groups/Listing Operations (CT)	1-6
Comments (AA)	1-7
Application Generation Run	1-8
Run Control Group (RC, RP, RF, ?T, IT)	1-9
File Definition Statements (FD, LS, LR, L0, Ln)	1-10
Screen/COMMAREA Control (SC)	1-10
Screen/COMMAREA Definition Statements (SD, SL, SS, S0, Sn)	1-11
Request Processing Statements (ER, TF, PR)	1-11
Catalog Retrieval Requests/Groups (CR)	1-12
Comments (AA)	1-13
Source Statement Retrieval Run	1-13
Run Control Group (RC, RP)	1-13
Catalog Retrieval (CT)	1-14
Library Utilities	1-14
Utility Control (UC)	1-14
Comments (AA)	1-15

Chapter 2: Statement Types

?T Statement - Execution Tracing Options	2-1
Run Name [1-8]	2-1
Statement Type [9-10]	2-1
Trace Control [11]	2-2
Application Flow Tracing Level [12]	2-3
Procedure Tracing [13]	2-3
Segment Action Tracing [14]	2-3
Intermediate Structure Loop Tracing [15]	2-4
DSDB Loop Tracing [16]	2-4
SSDB I/O Call Tracing Level [17]	2-4
DSDB I/O Call Tracing Level [18]	2-5
AA Statement - Comment	2-5
Name [1-8]	2-6
Statement Type [9-10]	2-6
Text [14-72]	2-6
CR Statement - Catalog Request	2-6
Cataloged Request Name [1-8]	2-6
Statement Type [9-10]	2-7
CT Statement - Catalog Listing/Retrieval	2-7
New Name or Group Name [1-8]	2-7
Statement Type [9-10]	2-7
Operation [11-13]	2-7
Item Names [14-61]	2-8
Continuation [62]	2-8
ER Statement - Establish Request	2-9
Request Name [1-8]	2-9
Statement Type [9-10]	2-9
Catalog Action [11]	2-9
Request Type [56]	2-9
Screen Name [57-64]	2-11
Back Branch Limit [65-69]	2-11
Reinitialize Temporary Fields? [70]	2-11

FD Statement - File Definition	2-12
File Name [1-8]	2-12
Statement Type [9-10]	2-12
File Identification [11-18]	2-12
DELETE? [19]	2-13
Glossary [20]	2-13
Record Format [21]	2-13
Record Size [22-25]	2-14
Buffer Size [29-33]	2-14
IT Statement - Item Tracking	2-15
Run Name [1-8]	2-15
Statement Type [9-10]	2-15
Generic Item Name? [11]	2-15
Type Code [12-13]	2-15
Item Name [14-21]	2-16
Expiration Date/Retention Period [30-35]	2-16
Updater ID [36-55]	2-16
Ln Statement - Comment	2-17
File Name [1-8]	2-17
Statement Type [9-10]	2-17
Segment/Field Name [11-18]	2-17
Delete? [19]	2-17
Comment [44-59]	2-17
LR Statement - Logical Relationship	2-18
File Name [1-8]	2-18
Statement Type [9-10]	2-18
Segment Name [11-18]	2-18
Delete? [19]	2-18
Sequence Number [20-22]	2-19
Logic Level [23]	2-19
Connector [24]	2-19
Relational Field Name [27-44]	2-19
Operation [45-46]	2-20
Qualifier [48]	2-20
Constant or Primary Key [49-66]	2-21

LS Statement - Segment Definition	2-22
File Name [1-8]	2-22
Statement Type [9-10]	2-22
Segment/Field Name [11-18]	2-22
Delete? [19]	2-22
Segment Number [20-21]	2-23
Suppress Duplication? [30]	2-23
Segment Order [31]	2-23
Creator ID or Authorization ID [36-43]	2-24
DB2/SQL External Name [44-61]	2-24
L0 Statement - Field Definition	2-25
File Name [1-8]	2-25
Statement Type [9-10]	2-25
Segment/Field Name [11-18]	2-25
Delete? [19]	2-25
Segment Number [20-21]	2-25
Level Number [22]	2-26
Field Location [23-26]	2-26
Field Length [27-29]	2-27
Field Type [30]	2-27
Segment Key [31]	2-29
Decimal Places [33]	2-29
Count Field For Segment [34-35]	2-30
This Segment Occurs N Times [36-38]	2-30
Output Edit/Length [42-43]	2-30
Alias [44-51]	2-31
DB2/SQL External Name [44-63]	2-31
PR Statement - Processing And Selection	2-32
Request Name [1-8]	2-32
Statement Type [9-10]	2-32
Statement Label [11-13]	2-33
Logic Level [14]	2-33
Connector [15]	2-33
Operand A [16-24]	2-34
Qualifier A [16]	2-34
Field Name A [17-24]	2-34
Operation [25-26]	2-35
Operand B [27-58]	2-37
Operand B Qualifier [27]	2-38
Field Name B or Constant [28-58]	2-38
Operand C [59-67]	2-40
Operand C Qualifier [59]	2-41

Operand C Field Name [60-67]	2-41
Partial Field [68-72]	2-41
Starting Character [68-69]	2-41
Number of Characters [70-71]	2-42
Operand A, B or C [72]	2-43
PR Statement - Call Option	2-43
Request Name [1-8]	2-43
Statement Type [9-10]	2-43
Statement Label [11-13]	2-43
Logic Level [14]	2-44
Call, PLI or DLI [15-18]	2-44
Load Module or CSECT Name [20-27]	2-44
Parameters	2-44
Continuation [72]	2-45
RC Statement - Run Control	2-46
Run Name/Transaction Code [1-8]	2-46
Statement Type [9-10]	2-46
Library Open [28]	2-47
System Delimiter [29]	2-47
Debug Level [30]	2-47
Source Statement Listing [31]	2-48
LKEDCALL Programs [32]	2-48
System Rollback Control [33]	2-49
CICSKEY Handling [34]	2-50
Message Processing [36]	2-50
Positional PCB? [37]	2-51
Run Type [38]	2-51
Target Monitor [39]	2-52
Target Operating System [40]	2-52
PSB Name [41-48]	2-52
Program Name [49-56]	2-53

RF Statement - Application Files	2-53
Run Name [1-8]	2-54
Statement Type [9-10]	2-54
File Name [11-18]	2-54
File Type [22-27]	2-55
Synchronizing Field 1 [28-35]	2-55
Synchronizing Field 2 [36-43]	2-56
Sync Qualifier [52]	2-56
Screen Sync Type [53]	2-57
DL/I Path Call Usage [54]	2-57
Record Length Field Name [55-62]	2-58
Storage Type [64]	2-58
Authorization ID [65-72]	2-59
RP Statement - Run Parameter	2-60
Run Name [1-8]	2-60
Statement Type [9-10]	2-60
Parameter Name [11-18]	2-60
Parameter Value [19-71]	2-61
SC Statement - Screen/COMMAREA Control	2-62
Input Screen or COMMAREA Section Name [1-8]	2-62
Statement Type [9-10]	2-62
Statement Sequence [11-13]	2-62
Logic Level [14]	2-63
Connector [15]	2-63
Operand A [16-24]	2-64
Qualifier A [16]	2-64
Field Name A [17-24]	2-64
Operation [25-26]	2-64
Operand B [27-49]	2-66
Operand B Qualifier [27]	2-67
Operand B [28-49]	2-68
Output Screen or COMMAREA Section Name [50-57]	2-70
Operand C [59-67]	2-70
Qualifier C [59]	2-70
Field Name/Constant [60-67]	2-71
Partial Field [68-72]	2-71
Starting Position [68-69]	2-71
Number of Characters [70-71]	2-72
Operand A, B, or C [72]	2-72

SD Statement - Screen/COMMAREA Definition	2-73
Screen or COMMAREA Section Name [1-8].....	2-73
Statement Type [9-10].....	2-73
Catalog Action [11].....	2-73
Glossary [12]	2-74
Field Designator [13]	2-74
Highlight Designator [14]	2-74
Automatic Format [15].....	2-75
Output Control [16]	2-76
Input Transaction Code [17-24].....	2-76
Screen Identifier [25-31]	2-77
Null Segment Processing [35]	2-77
Device ID [36-43].....	2-78
SL Statement - Screen/COMMAREA Layout	2-78
Screen Name [1-8].....	2-78
Statement Type [9-10].....	2-78
Line Number [11-13].....	2-79
Continuation [14]	2-79
XATTR [15].....	2-79
Text [16-72].....	2-80
Sn Statement - Screen/COMMAREA Content.....	2-82
Screen Name [1-8].....	2-82
Statement Type [9-10].....	2-82
Screen Segment Number [11-12].....	2-83
Virtual/Same/ After [15]	2-83
Operand A [16-24]	2-83
Action [38]	2-83
Logic Level [39].....	2-84
Connector [40].....	2-85
Edit/Validation Operator [41-42].....	2-85
Operand B [43-67].....	2-88
Qualifier B [43].....	2-88
Edit/Validation Field [44-67].....	2-89
EQ/NE - Pattern Validation.....	2-90
ED Operator - Edit Masks	2-91
EC/ET Operators - Error Condition Codes.....	2-92
Control Field Qualifier [43]	2-93
Control Field Name [44-51]	2-93
Control Operator [52-53]	2-93
Control Field Value Qualifier [54]	2-94
Control Field Value [55-67]	2-94
Partial Field [68-72].....	2-94

Starting Position [68-69]	2-94
Number of Characters [70-71]	2-94
Operand A, B, or S [72]	2-95
SS Statement - Structured Screen/COMMAREA Segment	2-96
Screen Name [1-8]	2-96
Statement Type [9-10]	2-96
Screen Segment Number [11-12]	2-96
Occurrences [14-17]	2-96
S0 Statement - Screen/COMMAREA Field	2-97
Screen Name [1-8]	2-98
S0 Statement Type [9-10]	2-98
Screen Segment Number [11-12]	2-98
Virtual/Same/ After [15]	2-98
Operand A, Source/Destination Field [16-24]	2-99
Qualifier A [16]	2-99
Operand A Field Name [17-24]]	2-99
Input/Output [25]	2-100
Input Required? [26]	2-100
Highlighting [27]	2-101
Cursor [28]	2-101
Error Handling [29]	2-102
Protected? [30]	2-102
Display Any Modification? [31]	2-102
Numeric [32]	2-103
Action [38]	2-104
Logic Level [39]	2-106
Connector [40]	2-106
Edit/Validation Operator [41-42]	2-107
Operand B [43-67]	2-108
Qualifier B [43]	2-109
Edit/Validation Field [44-67]	2-109
Partial Field [68-72]	2-110
Starting Position [68-69]	2-110
Number of Characters [70-71]	2-111
Operand A, B, or S [72]	2-111

TF Statement - Temporary Field	2-112
Request Name [1-8]	2-112
Statement Type [9-10]	2-112
Field Name [11-18]	2-112
Field Length [19-21]	2-113
Field Type [22]	2-113
Decimal Places [23]	2-114
Output Edit Length [27-28]	2-114
Initial Value [29-44]	2-114
Comments [45-72]	2-114
UC Statement - Utility Control	2-115
Run Name [1-8]	2-115
Statement Type [9-10]	2-115
Operation Code [11-14]	2-116
Generic Item Name? [15]	2-116
Item Type Code [16-17]	2-116
Item Name [18-25]	2-117
Library DD/DTF Name [34-41]	2-117
Replace Option? [42]	2-117
Replace Rule [43-45]	2-117
Directory Blocking Factor [46-49]	2-118
Purge Expired Items? [50]	2-118

Chapter 3: Advantage VISION:Transact Flags and Patterns

VISION:Transact Flags	3-1
Today, Date, ISDATE, and Julian Flags.	3-3
Time Flag.	3-4
CSTATUS and RETURNCD Flags.	3-4
ATTNID Flag	3-6
ERROR Flag	3-7
TRANCODE Flag	3-7
PROGTYP Flag.	3-8
SYNCOK Flag	3-8
GOSC Flag	3-8
TERMID, USERID, OPERID, and OPERCL Flags	3-8
MSGLINE FLAG	3-10
Dynamic Partial Field Flags	3-10
HIGHVALU and LOWVALU Flags	3-11
PAGENO Flag	3-11
SCRNAMEI and SCRNAMEO Flags	3-12
Execution Tracing Flags	3-13
INVFIELD Flag	3-15
Validation Patterns	3-15
Named Patterns	3-15

Index

Introduction

Overview

Advantage™ VISION:Transact® is an implementation system for developing online business applications using a high level language. These applications run in the IMS TM or CICS TS environment.

The VISION:Transact system consists of two major components: the VISION:Transact generator and the VISION:Transact executor. The generator, using the statements described in this guide, creates online applications, while the executor is used online by the generated application.

The VISION:Transact generator has four functions: definition (catalog maintenance), application generation, source statement retrieval, and library utilities. Each function requires a specific run type that determines what statements are acceptable. [Figure 1-1 on page 1-2](#) shows the various statement types acceptable for each run type.

Note: Beginning with this release, the CICS TS version of VISION:Transact supports COMMAREA programs in addition to 3270 Terminal programs. The VISION:Transact statements used to define 3270 screen layout and content are also used to define COMMAREA layout and content. Unless specifically noted otherwise, any reference to the term ‘screen’ hereafter in this document is assumed to also mean ‘COMMAREA.’

The VISION:Transact generator requires use of the COMLIB subsystem to perform definition, application generation, source statement retrieval, and library utility runs.

Definition	Application Generation	Source Statement Retrieval	Library Utility
Run Control Group RC RP IT	Run Control Group RC RP RF ?T IT	Run Control Group RC RP	Utility Control UC
File Definition Statements FD LS LR L0 Ln	File Definition Statements FD LS LR L0 Ln	Catalog Retrieval CT	Comments AA
	Screen/COMMAREA Control SC	Comments AA	
Screen/COMMAREA Definition Statements SD SL SS S0 Sn	Screen/COMMAREA Definition Statements SD SL SS S0 Sn		
Request(s) ER TF PR	Request(s) ER TF PR	Note: Procedures may also be input during an application generation run. The syntax for procedures is documented in the <i>Advantage VISION:Transact ASL Reference Guide</i> .	

Figure 1-1 Run Types and Statement Sequence (Page 1 of 2)

Definition	Application Generation	Source Statement Retrieval	Library Utility
Catalog Request Group(s)/ Listing Operations CT	Catalog Retrieval Request(s)/ Group(s) CR		
Comments AA	Comments AA		

Note:

The Definition, Application Generation, and Source Statement Retrieval runtypes executes program MARKV or M5.

The Library Utility run type executes program M5UTIL.

Figure 1-1 Run Types and Statement Sequence (Page 2 of 2)

Not all the statements are necessarily required nor are they handled in the same way for all run types. While some statements are acceptable in different runs, how VISION:Transact processes these statements varies depending on the run type. For example, file definition statements included in a definition run are cataloged on the primary VISION:Transact library, while file definition statements included in an application generation run are not. The following section describes each of the four run types and how each statement group functions within its run type. For specific entries of each statement type refer to the alphabetic listing following this section.

Definition Run (Catalog Maintenance)

The purpose of a definition run is to create, update, replace, or delete items in an existing library or to list the items on the library. The types of items are file and screen definitions, requests, and request groups. Once items are cataloged or stored in the library, they may be retrieved for use in an application generation run. In a definition run, file definitions may be created, updated, replaced, or deleted; screen definitions may be created, replaced, or deleted; individual requests may be created, replaced, or deleted; request groups may be created or deleted. You also have the option of listing the names of all items of a given type or the contents of all items or selected items. Only the primary VISION:Transact library is updated in a definition run. Alternate libraries are only searched in a definition run when CT operations name specific items to be listed.

Run Control Group(RC, RP, IT)

In a definition run, the first input statement must be a run control (RC) statement that identifies the run type (D or blank for a definition run) and the teleprocessing monitor. The teleprocessing monitor entry (I or C) is used to determine whether BMS or MFS statements are output for any screen definitions included in the definition run. Included in the run group (immediately following the RC) may be run parameter (RP) statements for multiple VISION:Transact libraries (used for overriding the primary library, M5LIB). Item tracking (IT) statements may also be included for assigning expiration date/retention period data to items updated or created by the run on the library as well as data to identify the creator or updater of the item.

File Definition Statements (FD, LS, LR, L0, Ln)

Including file definition statements in a definition run results in either creating, updating, replacing, or deleting file definitions on the primary VISION:Transact library. File definitions for IMS, fast path, and DB2 databases and VSAM/ISAM files, working storage, pseudo files and PCB masks may be included and are cataloged, updated, replaced, or deleted accordingly. Updates to an existing file definition may be limited to a single statement update. For example, you may update a particular L0 statement (for a field) within a file definition. You may also update multiple statement types in a single file definition. It is suggested that an update of this type be done by replacing the entire file definition using the FD delete entry of R, followed by the new as well as the unaltered statement types for the entire file definition.

A file definition is only placed in the library if it is valid. Therefore, if you attempt to update, replace, or create a file definition and any statement is invalid, the update to the library does not take place. In the case of updating or replacing, the original version of the file definition remains on the VISION:Transact library.

Screen/COMMAREA Definition Statements (SD, SL, SS, S0, Sn)

In a definition run, you may include screen definition statements that create a new screen definition, replace an existing screen definition, or delete an existing one on the library. However, you cannot update or input a screen definition that already exists in the library. For example, updating a single statement in an existing screen definition is not allowed. In order to update a screen definition, you must either replace the entire screen definition with a new version or delete the old version of the screen (using an SD statement) and input all screen definition statements for the new version of the screen.

Screen definitions cataloged in a definition run or APPGEN run are cataloged to the library prior to any validation. In other words, it is possible to catalog a screen definition with errors. In a definition run, VISION:Transact first catalogs the screen definition in the library and then checks screen definitions for syntax errors; screen definitions are cataloged in the VISION:Transact library regardless of the type of error. (Additional validation takes place in the application generation run. Thus, a screen definition that is valid during a definition run can become invalid during APPGEN.) In order to correct a screen definition, you must either replace it or delete the incorrect version from the library (with an SD statement) and input the corrected version. MFS or BMS source statements are automatically generated and listed for valid screen definitions.

Request Processing Statements (ER, TF, PR)

Individual requests may be created, deleted, or replaced on the library in a definition run. Requests included in a definition run are added to the library; this is the default. If the request already exists on the library and you specify that the request be added by the ER statement catalog action entry, an error message is issued and the library is not updated. Updating a request may be accomplished two ways. You may delete the existing version (using an ER statement) and input the new version, or you may specify that the existing version be replaced. Specifying a replace catalog action is similar to the delete and add; however, it only requires a single ER statement. The request is replaced if it already exists on the library; if not, it is added to the library. Deleting a request that happens to belong to a request group does not remove the request name from the request group. If the request group is used, an error is issued indicating that the request is not in the library.

Requests, like screen definitions, are cataloged to the library prior to any validation. If a request is cataloged with errors, it must be replaced with the corrected version. In addition, if a request is to be referenced via an XR or OS operator in screen control (SC) statements within an application generation run, the screen name entry on the ER statement must be blank.

Catalog Request Groups/Listing Operations (CT)

You may create or delete an existing request group in a definition run with the CT statement SAV and DEL operations. Updating an existing request group (adding requests to a request group) is not allowed. In order to update a request group, you must first delete the existing request group. You can then save the request group containing any new requests as well as the existing requests with the original request group name. When adding a request group to the library, it is essential that each type 1, type 2, type M, and type O request have a screen name in its ER statement. Recall that requests referenced in screen control statements via the XR and OS operators within an application generation run must not have a screen name entry. For this reason, requests referenced by an XR or OS operator must not be cataloged in a request group.

In addition, request groups are cataloged prior to validation.

Deleting an individual request from the library does not cause the request to be removed from the request group; if the request group is used, VISION:Transact issues an error that the request is not on the library. To remove a request from a group, the request group must first be deleted and then saved with the desired requests.

The listing operations supported in a definition run include the following:

For Definition Listing Operations

LFN	List the names of all file definitions.
LSN	List the names of all screen definitions.
LIN	List all item names.
LFG	List file definition glossaries.
LSG	List screen definition glossaries.
LFV	List file definition view glossaries.

For Cataloged Request Operations

LST	List the names of all request(s) and request group(s).
DMP	List the contents of requests.
DEL	Delete a request group.
SAV	Save a request group.

Comments (AA)

Comments may be included throughout the definition run with the following restrictions:

1. AA statements may not appear ahead of the Run Control (RC) statement.
2. AA statements must not separate a logical expression; they may only appear before or after a complete logical expression (S0/Sn, SC, PR).
3. AA statements must not separate screen definition (S0/Sn) statements that form a dynamic action group (IF operator).
4. AA statements must not separate screen definition (S0/Sn) statements that have EC and ET operators.
5. AA statements included within screen layout (SL) statements will be sorted by VISION:Transact and appear on the source listing after the SD statement and before all SL statements.

Comments are not listed in definition glossaries.

Application Generation Run

The purpose of an application generation (APPGEN) run is to generate an application for a particular teleprocessing environment. All elements required for a complete application (file definitions, screen definitions, requests, procedures, and screen control) may be included.

Note: Procedures can be included in an APPGEN run but are only for temporary use during the APPGEN; they cannot be placed in the library. Procedures are documented in the *Advantage VISION:Transact ASL Reference Guide*.

The LIBOPNA keyword in M5PARM specifies whether the primary library should be opened for update access or read only access during an application generation run. With read only access, screen definitions cannot be input and individual requests may be input but cannot be added, replaced, or deleted on the library in an APPGEN run. The Library Open [28] entry on the RC statement can override the specification in M5PARM.

The run control group must be the first group of statements in an APPGEN followed by all instream file definition statements. After the file definition statements may be screen definition statements, requests, procedures, screen control statements, etc., in any order.

In other words, file definitions may be followed by requests followed by screen control. However, the statements for a single item must be grouped together. For example, you cannot follow screen definition statements by a request and then include additional screen definition statements for the same screen.

If XR and OS operators in screen control reference requests not included in the input stream, VISION:Transact automatically attempts to retrieve the requests from a VISION:Transact library. The same is true for type S requests referenced but not included in the input stream. If a request is referenced and is not found on the library or the input stream, an error is issued and the application cannot be generated.

Although request groups cannot be cataloged in an APPGEN, they may be retrieved from the library for inclusion in the application via a CR statement. In addition, any file definition statements included in an APPGEN only create, replace, or update the file definition for that particular application; file definitions are not created, replaced, or updated on the library in an APPGEN. The only update functions to the library supported in an APPGEN are those for screen definitions and individual requests. These update functions are only allowed when the library is opened for update (see the library open specification in M5PARM and on the RC statement).

If no type 3 or higher errors are detected in the APPGEN, the object code for the application (written to the M5PUNCH data set) as well as the recommended TP monitor statements, are generated and printed on the VISION:Transact source listing for the application. In addition, the TP monitor statements are written to M5TPTBL provided a DD/DTF statement is included for that data set in the APPGEN JCL. Recommended TP monitor statements for IMS TM include the PSB, APPLCTN, and TRANSACT statements. TP monitor statements for CICS TS include the PCT and PPT entries, PSB if IMS databases are used, and RCT if DB2 tables are used.

If screen definitions were included during an APPGEN run, the MFS/BMS source statements are written to the M5MAP data set.

Note: CICS TS PPT table entries for screen definitions used in the application but cataloged previously in a definition run are not included in the M5TPTBL data set. In other words, PPT table entries are written to M5TPTBL only for screen definitions included in the APPGEN.

Run Control Group (RC, RP, RF, ?T, IT)

In an APPGEN run, the first statement in the run control group must be the run control (RC) statement identifying the run as an APPGEN (run type G). Following the RC statement are any run parameter (RP), run file (RF), execution tracing options (?T), or item tracking (IT) statements.

The RC statement for an APPGEN run is used to assign the transaction code, program name, and PSB name to be used by the application as well as target monitor and special processing options to be used by the application.

The RP statement(s) serve four functions:

1. Specify the primary and alternate libraries to be used by the application.
2. Generate design manual documentation.
3. Specify generation of a fast path application (IMS TM only).
4. Specify specific called programs that are to be link edited with the generated application.

RP MULTILIB statements specify how multiple libraries are used in the APPGEN. Although all libraries are searched during retrieval operations, only the primary library is used in update operations.

The RF statements are used to identify the databases, working storage, pseudo files, or PCB masks to be used by the application and to name their associated file definitions. You may generate an application without RF statements. This is useful for menu transactions where no files are necessary or for prototyping screen flow. For example, you can develop the screens and screen control for an application without having to define the files or identify them to the application.

The ?T statement is used to include or exclude execution tracing within an application and establish defaults to be used for the individual tracing options when application execution begins. The execution tracing options available are application flow, procedure, segment actions, looping, and I/O calls.

IT statements may be included to provide expiration date/retention period data of items included in the APPGEN as well as creator/updater ID information. However, IT statements are applicable only to those items created, replaced, or updated in the library. Because file definitions are not created, replaced, or updated in the library during an APPGEN run, IT statements included in file definitions are not applied.

File Definition Statements (FD, LS, LR, LO, Ln)

File definition statements included in an APPGEN do not update the primary library. In order to update, create, replace, or delete a file definition in the library, the file definition statements must be input to a definition run.

If you input file definition statements for an item that does not exist on the library, the instream statements create a new file definition for use only in that APPGEN run. If the file definition is invalid, the application is not generated.

If you input file definition statements for an item that exists on the library, the instream statements either modify the file definition or replace the entire file definition for that application. Modifying a file definition causes retrieval of a cataloged file definition to which modifications are made only for that application. Replacing a file definition only occurs when the FD statement Delete entry contains an "R" and the instream file definition exists on the library. When a file definition is replaced during an APPGEN run, the cataloged definition is completely overridden and a new file definition is temporarily created using the instream statements. If there is an error in the new version of the file definition, the application is not generated.

Screen/COMMAREA Control (SC)

Screen/COMMAREA control statements are included in an APPGEN to specify screen-to-screen or COMMAREA section-to-COMMAREA section flow. You may also use screen/COMMAREA control statements to name instream requests or procedures or cataloged requests (using the XR and OS operators) that are to be processed for input and output screens or input and output COMMAREA sections. Remember that screen/COMMAREA control statements are not sorted by statement sequence; they are processed in input sequence.

After all input is read, the screen control statements are validated and listed. Each screen definition named in screen control is then retrieved and validated. If a screen definition is included in the input stream and is not referenced on a SC statement, an error is issued and the application is not generated. BMS or MFS source statements are generated for those valid screens included in the input stream. (A DD/DTF statement for M5MAP must be included for the generated source.)

Screen/COMMAREA Definition Statements (SD, SL, SS, SO, Sn)

***Important!** The term 'screen' below also implies 'COMMAREA section.'*

Screen definition statements may be included in an APPGEN to create, replace, or delete screen definitions on the primary library depending on the library open specification in M5PARM and on the RC statement. Instream statements to create or replace a screen definition are cataloged in the primary library prior to validation. If you attempt to create a screen definition which is already in the library, an error is issued and the screen definition statements are rejected. In order to update an existing screen definition, you must either replace the entire definition or first delete the original version (using an SD statement) and input the new version with the same screen name. In addition, if you input a screen definition and subsequently attempt to delete the same definition, an error is issued and the delete is ignored.

Note: After all statement types have been read and screen control has been validated, VISION:Transact validates the screen definitions used in the application. All screen definitions in the input stream must be referenced by a screen control statement. If not, an error is issued for each screen definition. VISION:Transact attempts to retrieve from the primary library, and then the alternate libraries, any screen definitions named in screen control but not included as input to the APPGEN. If the screen definitions input to an APPGEN are valid, MFS or BMS source statements are generated for those screens. An M5MAP DD/DTF statement must be provided for these source statements.

Request Processing Statements (ER, TF, PR)

Request statements included in an APPGEN may be used to specify requests that are for temporary use in the APPGEN. Requests may be added, deleted, or replaced in the primary library during APPGEN if the primary library is opened for update access (see the library open specification in M5PARM and on the RC statement). The default action for requests included in an APPGEN run is not to catalog them to the library. If you specify that a request is to be added or replaced (using the catalog action entry on the ER statement), then the request is added or replaced prior to validation. If the request to be added or replaced is already in the library, an error is issued and the request is not cataloged. Remember that requests referenced through XR or OS operators (screen control statements) must not have a screen name entry on the ER statement.

VISION:Transact pre-processes all TF statements in all requests used in an APPGEN and also identifies the names of the type S requests (subroutines) invoked by requests. Explicitly defined TF statements are processed first to guarantee that the explicit definitions take precedence over implicit definitions if implicit definitions are allowed in the VISION:Transact system (see the TFREQ specification in M5PARM). In other words, before creating an implicit temporary field, VISION:Transact ensures that it is not explicitly defined elsewhere in the run. VISION:Transact ignores duplicate temporary field statements and uses the first reference for its definition. If a type S request is referenced but is not in the input stream, VISION:Transact attempts to retrieve it from a VISION:Transact library. If the subroutine is not on a library, an error is issued.

If requests named on XR or OS operators in screen control are not found in the APPGEN input, VISION:Transact will attempt to retrieve them from the library. If they are not on the library, an error is issued.

Catalog Retrieval Requests/Groups (CR)

If catalog retrieval (CR) statements are included in an APPGEN, the named request or request group is retrieved from a VISION:Transact library. If multiple libraries are specified, the libraries are searched (in order) until the request or request group is found. An error is issued if catalog retrieval statements name a request previously referenced in the input stream.

In addition, if a catalog retrieval statement references an individual request that cannot be found in a VISION:Transact library, an error is issued. Requests retrieved by CR statements are also pre-processed for TF statements and type S requests.

Remember that requests cataloged in a request group must have a screen name in the ER statement and therefore cannot be referenced by XR or OS operators in screen control.

There are three types of processing involved when a request group is retrieved:

1. If a request in the group has the same name as one that has been previously processed, then that request is used and its location in the processing sequence is determined by its location in the request group.
2. If the request in the group is not included in the input stream, VISION:Transact attempts to retrieve it from a VISION:Transact library.
3. If a request in a request group is not found, an error is issued.

Comments (AA)

Comments may be included throughout an APPGEN with the following restrictions:

1. AA statements may not appear ahead of the run control (RC) statement.
2. AA statements must not separate a logical expression; they may only appear before or after a complete logical expression (S0/Sn, SC, PR).
3. AA statements must not separate screen definition (S0/Sn) statements that form a dynamic action group (IF operators).
4. AA statements must not separate screen definition (S0/Sn) statements that have EC and ET operators.
5. AA statements in screen control (SC) must not separate statements that have OS and OC operators.
6. AA statements included within screen layout (SL) statements will be sorted by VISION:Transact and appear on the source listing after the SD statement and before all SL statements.
7. AA statements used in screen control (SC) must be blank in positions [1-8].

Comments are not listed in definition glossaries.

Source Statement Retrieval Run

A source statement retrieval (SSR) run is used to retrieve items from a VISION:Transact library and output the items in source statement form. Statements output from a source statement retrieval run are written to the M5SSOUT data set. In addition, other statement types may be included in a source statement retrieval run and are output directly from M5INPUT to M5SSOUT. The M5SSOUT data set may then be used as input to a subsequent VISION:Transact run.

Run Control Group (RC, RP)

The first input statement must be the run control (RC) statement identifying the run as a source statement retrieval run (Run Type of S). Multiple libraries may be specified in a source statement retrieval run (through JCL and RP MULTILIB statements), however, the libraries searched depend on the CT retrieval operation used. For example, if a specific screen definition is requested, all available libraries are searched for that item until it is found. Only the first occurrence of the item is output. If all items of a given type are requested (for example, file definitions), then only the primary library is searched.

Catalog Retrieval (CT)

CT statements are included in a source statement retrieval run to retrieve source statements of different items on a VISION:Transact library. The following CT statements are supported in a source statement retrieval run:

RFD	Retrieve file definitions.
RSD	Retrieve screen definitions.
RRG	Retrieve requests or request groups.
EOF	Output a /* statement to M5SSOUT.

Comments may be included in a source statement retrieval run but are treated as input statements to be written to M5SSOUT. Other statements input to a source statement retrieval run (other than the first RC, RP MULTILIB statements and CT retrieval statements) are copied directly to M5SSOUT.

Note: APPGEN source is not kept.

Library Utilities

The utility program (M5UTIL) is used to create and maintain VISION:Transact libraries. All functions of the library utilities may be performed using the utility control (UC) statements and multiple utility control statements may be used in a single utility run. Only UC statements and comments (AA statements) can be input to the M5UTIL program; an RC statement is not allowed.

Utility Control (UC)

The utility control (UC) statement is used to specify which operation to perform in the utility run. The following are the operation codes that may be specified in a utility run:

INIT	Initializes libraries.
DUMP	Creates a backup copy of a library.
REST	Restores a library from a backup copy.
COND	Condenses or reorganizes a library.
COPY	Copies items from one library to another.
MERG	Copies all of the items in a library to another library.

Comments (AA)

Comments may be included in utility run.

Statement Types

?T Statement - Execution Tracing Options

In an application generation run only, use a ?T statement to include or exclude execution tracing within an application. A ?T statement also specifies the defaults to be used for the individual tracing options when application execution begins. All trace options should be specified on a single ?T statement. If multiple ?T statements are input, the last one encountered determines the trace options used.

Run Name (1-8)

Required entry

Enter the name specified in positions [1-8] of the RC statement.

Statement Type (9-10)

Required entry

Enter ?T.

Trace Control (11)

Use this entry to generate an application with or without execution tracing. Enter one of the following codes:

Entry	Result
P	Generate the application without tracing as it is to be placed into production. This entry is equivalent to supplying no ?T statement. With this entry, code is <i>not</i> generated into an application for the processing of trace operators or commands used in requests or procedures. Thus, trace operators or commands do not have to be removed from requests or procedures since they do not affect the size of the generated application load module. Note: With this entry, the trace flags cannot be used to enable or disable tracing; any modifications to the trace flags are ignored.
Blank/0	Generate the application with tracing and begin application execution with all tracing disabled.
1	Generate the application with tracing and begin application execution with individual traces enabled or disabled based on the contents of their corresponding ?T statement entry.

The Trace Control entry is used to initialize flag TCONTROL at the beginning of application execution. If Trace Control contains a 1, flag TCONTROL is initialized to 1. In all other cases, flag TCONTROL is initialized to 0. When an application is generated with tracing (Trace Control entry of 0 or 1), the trace flags may be used to enable or disable all or individual traces.

Note: For CICS applications, the trace options in effect are retained between interactions until the pseudo conversation terminates which can span multiple transactions. Thus, the ?T statement entries from the transaction that initiates the pseudo conversation are used to initialize the trace flags. Subsequent transactions of the pseudo conversation do not reinitialize the trace flags with the ?T statement entries input during their respective APPGENs.

Application Flow Tracing Level (12)

This entry specifies the application flow trace level to be used when application execution begins. The value specified here is used to initialize flag TAPPFLOW and may be overridden by changing the contents of flag TAPPFLOW. Enter one of the following codes:

Entry	Result
Blank/0	Indicates trace is disabled.
1	Indicates trace is enabled at Level 1; only Level 1 application flow messages (AFnn) and teleprocessing (TP) interface calls for screen and saved working storage/pseudo files (TPnn messages) are output. The I/O areas for screens and saved working storage/pseudo files are not displayed.
2	Indicates trace is enabled at Level 2; Level 1 and 2 application flow messages are output as well as TP interface calls and I/O areas for screens and saved working storage/pseudo files.

Procedure Tracing (13)

This entry specifies whether procedure (request) tracing is enabled or disabled when application execution begins. The value specified here is used to initialize flag TPROC and may be overridden by changing the contents of flag TPROC. Enter one of the following codes:

Entry	Result
Blank/0	Indicates trace is disabled.
1	Indicates trace is enabled which also enables the execution of tracing operators or commands within a request or procedure.

Segment Action Tracing (14)

This entry specifies whether segment action tracing is enabled or disabled when application execution begins. The value specified here is used to initialize flag TSEGACT and may be overridden by changing the contents of flag TSEGACT. Enter one of the following codes:

Entry	Result
Blank/0	Indicates trace is disabled.
1	Indicates trace is enabled.

Intermediate Structure Loop Tracing (15)

This entry specifies whether intermediate structure loop tracing is enabled or disabled when application execution begins. The value specified here is used to initialize flag TILOOP and may be overridden by changing the contents of flag TILOOP. Enter one of the following codes:

Entry	Result
Blank/0	Indicates trace is disabled.
1	Indicates trace is enabled.

DSDB Loop Tracing (16)

This entry specifies whether DSDB (dynamically synchronized database) loop tracing is enabled or disabled when application execution begins. The value specified here is used to initialize flag TDLOOP and may be overridden by changing the contents of flag TDLOOP. Enter one of the following codes:

Entry	Result
Blank/0	Indicates trace is disabled.
1	Indicates trace is enabled.

SSDB I/O Call Tracing Level (17)

This entry specifies the SSDB (screen synchronized database) I/O call tracing level to be used; loop tracing is enabled when application execution begins. The value specified here is used to initialize flag TSSDBIO and may be overridden by changing the contents of flag TSSDBIO. Enter one of the following codes:

Entry	Result
Blank/0	Indicates trace is disabled.
1	Indicates trace is enabled at Level 1 which outputs the calls but does not output the I/O areas.
2	Indicates trace is enabled at Level 2. Same as Level 1 except the I/O areas are also displayed.

DSDB I/O Call Tracing Level (18)

This entry specifies the DSDB (dynamically synchronized database) I/O call tracing level to be used when application execution begins. The value specified here is used to initialize flag TDSDBIO and may be overridden by changing the contents of flag TDSDBIO. Enter one of the following codes:

Entry	Result
Blank/0	Indicates trace is disabled.
1	Indicates trace is enabled at Level 1 which outputs the calls but does not output the I/O areas.
2	Indicates trace is enabled at Level 2. Same as Level 1 except the I/O areas are also displayed.

AA Statement - Comment

Comments may be included throughout an APPGEN with the following restrictions:

1. AA statements may not appear ahead of the run control (RC) statement.
2. AA statements must not separate a logical expression; they may only appear before or after a complete logical expression (S0/Sn, SC, PR).
3. AA statements must not separate screen definition (S0/Sn) statements that form a dynamic action group (IF operators).
4. AA statements must not separate screen definition (S0/Sn) statements that have EC and ET operators.
5. AA statements in screen control (SC) must not separate statements that have OS and OC operators.
6. AA statements included within screen layout (SL) statements will be sorted by VISION:Transact and appear on the source listing after the SD statement and before all SL statements.
7. AA statements used in screen control (SC) must be blank in positions [1-8].

Comments are not listed in definition glossaries.

Name (1-8)

Depending on where you use the comment, name it as follows:

Area	Naming Convention
Run control	Run control name.
Screen definition	Screen name.
File definition	File name.
Screen control	Leave blank.
Request	Request name.
Library utility	Leave blank.

Statement Type (9-10)

Required entry

Enter AA.

Text (14-72)

Enter the text for the comment as it is to appear in the source statement listing. All characters including the delimiter are valid.

CR Statement - Catalog Request

Use the CR statement in an application generation run to retrieve a request or request group that is cataloged in the VISION:Transact library and include it in an application.

Cataloged Request Name (1-8)

Required entry

Enter the name of the cataloged request or request group to be retrieved for this application.

Statement Type (9-10)

Required entry

Enter CR.

CT Statement - Catalog Listing/Retrieval

The CT statement can be included in a definition run or a source statement retrieval run. In a definition run, a CT statement can be used to list cataloged item names, definition glossaries, or the contents of cataloged requests. CT statements in a definition run also can be used to save and delete request groups. In a source statement retrieval run, a CT statement can be used to specify items to be retrieved.

New Name or Group Name (1-8)

In a definition run, use this entry to name a request group to be cataloged, otherwise leave it blank. In a source statement retrieval run, use this entry to rename a retrieved file or screen definition or request on the M5SSOUT data set.

Statement Type (9-10)

Required entry

Enter CT.

Operation (11-13)

In a definition run, enter one of the following operations:

LSN	List all screen definition names in the library.
LSG	List all or selected screen definition glossaries in the library. (The glossary format depends on the glossary type of the cataloged definition.)
LFN	List all file definition names in the library.
LFG	List all or selected file definition glossaries (the glossary format depends on the glossary type of the cataloged definition).
LFV	List all or selected file definition view glossaries in the library.
LIN	List all item names in the library.

LST	List the names of all requests and request groups in the library.
DMP	List the contents of all or selected requests/request groups in the library.
DEL	Delete one or more request groups from the library. The request groups are named in positions [14-61].
SAV	Save the requests in the order they are named in positions [14-61] under the request group named in positions [1-8].

In a source statement retrieval run, enter one of the following operations:

RFD	Retrieve all or selected file definitions from the library.
RRG	Retrieve all or selected requests/request groups from the library.
RSD	Retrieve all or selected screen definitions.
Note: Enter no item names to retrieve the source of all file definitions, screen definitions, or requests/request groups. Enter item names to retrieve the source of those items only. To rename a retrieved item, name only one item.	
EOF	To request that an end of file statement (/*) be output to M5PUNCH.

Item Names (14-61)

Leave blank for operations LSN, LFN, LIN, LST, and EOF. For the SAV operation, name one to six requests to be cataloged in the request group named in positions [1-8]. For the DEL operation, enter one to six request groups to be deleted.

For all other operations, name one to six specific items to list or retrieve. To rename a retrieved item (operations RFD, RSD, or RRG), name one item per CT statement in positions [14-21] and enter a new name in positions [1-8]. If positions [14-61] are blank, all item glossaries or contents are listed or retrieved.

Continuation (62)

To catalog more than six requests in a request group or delete more than six request groups, enter a non-blank character. Enter no operation code on the next CT statement and continue naming requests for the SAV operation or request groups for the DEL operation under Item Names [14-61]. This specification can also be used to continue a list of items to be retrieved or listed. In any case, enter no operation code in the next CT statement and continue naming items under Item Names.

ER Statement - Establish Request

The ER statement can be used in a definition run or an application generation run as the first statement of a request. Its specifications pertain to the request as a whole. The ER statement can also be used to direct the cataloging of a request into the library or to delete a request from the library.

Request Name (1-8)

Required entry

Enter a request name unique to this application.

Statement Type (9-10)

Required entry

Enter ER.

Catalog Action (11)

To catalog, replace, or delete a request from the library, enter one of the following codes:

Entry	Action
Blank or N	Do not catalog the request. This is the default in an APPGEN run.
D or Y	Delete the request from the library.
A	Add the request to the library if it is not yet cataloged. This is the default in a definition run.
R	Replace the request if it is already cataloged; otherwise add it.

In an APPGEN run, the library must be opened for update if a catalog action of D, Y, A, or R is used (see [Library Open \[28\] on page 2-47](#) for the Library Open specification in M5PARM and on the RC statement).

Request Type (56)

Required entry

Enter one of the following codes to specify the point in the interaction at which this request is to be executed:

Entry	Timing and Characteristics
Blank/N	Do not execute the request. The request is used to define temporary fields or provide comments only.
O	Execute the request after the database is physically updated, and the output screen is selected. No more than one type O request is executed per interaction in conjunction with the formatting of the output screen. A type O request may use all valid qualifiers except S.
1	Execute the request after the associated screen is input and de-editing has been performed for screen fields but before any screen data is validated. A type 1 request may reference data fields using qualifiers S, T, W, F, and V. Qualifiers blank/N and 1-9 may be used only if they are defined on the RF statement as dynamically synchronized files or pseudo files. Multiple type 1 requests are allowed.
2	Execute the request after screen data is validated, defaults are inserted, and segment actions concerning retrievals are performed. A type 2 request may use all valid qualifiers. Segments to be inserted by the screen are yet unavailable, and segments deleted by the screen are still available. Multiple type 2 requests are allowed.
M	Execute the request after all segment and field actions are completed on the intermediate structure, but before the database is physically updated. A type M request may use all valid qualifiers. Segments inserted by the screen are now available, and those deleted by the screen are no longer available. Multiple type M requests are allowed.
S	Execute the request only if it is invoked by another request. A type S request may use any qualifier that is available to its calling request.

Chapter 7 of the *Advantage VISION:Transact Reference Guide* discusses requests and their role in the interaction.

Screen Name (57-64)

Enter the name of the screen to which the request applies. A request can be assigned to a screen as the application is generated. In this case, this entry must be left blank and screen control must address the request with an XR or OS operator.

Entry	Result
1, 2, M	Enter the name of the associated input screen.
O	Enter the name of the associated output screen.
Blank/N/S	Leave this entry blank.

Back Branch Limit (65-69)

If the request includes a PR statement with a GO/GS/NS BACK branch, enter a number from 1 to 99999 to limit the number of back branches caused by the PR statement.

Reinitialize Temporary Fields? (70)

Enter one of the following codes:

Entry	Result
Blank/N	Do not reinitialize temporary fields when this request is entered.
Y	Reinitialize temporary fields defined on TF statements in this request when this request is entered but not on a back branch.
B	Reinitialize temporary fields defined on TF statements in this request, even on a back branch.

Note: Temporary fields are always initialized at the beginning of each interaction unless the RECYCLE parameter on the screen control OC statement is specified. For CICS TS only, temporary fields are not initialized at the beginning of conversational interaction (CONV parameter of the screen control OC statement).

FD Statement - File Definition

The FD statement can be used in a definition run or an application generation run as the first statement of a file definition. File definition statements included in an APPGEN only create, replace, or update the file definition for that application; they do not update the library.

File Name (1-8)

Required entry

Enter the name of the file as it is defined to VISION:Transact.

Statement Type (9-10)

Required entry

Enter FD.

File Identification (11-18)

When defining an IMS or fast path database, enter the physical DBD name for the database view. If positional PCB selection is specified in the APPGEN, this entry need not specify the DBD name.

Note: Refer to Chapters 3 and 10 of the *Advantage VISION:Transact Reference Guide* for PCB and DBD specifications.

For SQL files, pseudo files, working storage, or PCB masks, enter a comment or leave this entry blank.

When defining a VSAM or ISAM file, enter the File Control Table (FCT) name used by CICS to process this file.

DELETE? (19)

To create, replace, or delete a file definition from the library during a definition run or temporarily during an APPGEN run, enter one of the following codes:

Entry	Result
Blank/N	In a definition run, add the file definition to the library. In an APPGEN run, temporarily create a new file definition for use only in that APPGEN run.
R	In a definition run, replace the file definition if it is already cataloged on the library; otherwise add it to the library. In an APPGEN run, temporarily create a new file definition for use only in that APPGEN regardless of whether the file definition exists on the library.
Y	In a definition run, delete the existing file definition from the library. In an APPGEN run, this entry is invalid.

Glossary (20)

Enter one of the following codes to control the listing of the file definition contents:

Entry	Result
Blank/A	Besides file and record characteristics, the fields in each segment are listed alphabetically by field name.
B	Same as A, except the fields in each segment are listed by field location.
1	Produce an abbreviated glossary in which fields are listed alphabetically by field name within each segment.
N	Suppress the glossary.

The first page of a glossary lists the data attributes and the fields in the first segment. The data and field specifications of subsequent segments appear on their own page.

Glossaries of existing library items may be printed on any subsequent definition run. When requesting a glossary, specify only the file name, statement type, and glossary.

Record Format (21)

Required entry

Enter one of the following codes to identify the physical record format and access method:

Entry	Physical Record Format/Access Method
D	IMS database (not fast path).
1	IMS main storage database (fast path). May only have a root segment.
2	IMS data entry database (fast path). May have at most one sequential dependent.
F	Fixed length pseudo file, working storage area, or PCB mask.
V	Variable length pseudo file.
I	Fixed length ISAM file (ISAM files were desupported in CICS release 1.7).
J	Variable length ISAM file (ISAM files were desupported in CICS release 1.7).
K	Key-sequenced VSAM file (KSDS).
E	Entry-sequenced VSAM file (ESDS).
R	Relative record VSAM file (RRDS).
S	DB2 or SQL/DS table generation.

Record Size (22-25)

If the record format is F or I, enter the fixed record size in bytes. This entry is used with pseudo files, working storage, PCB masks, and fixed length ISAM files. The value specified must not exceed 9999. If this entry is left blank, a buffer size value should be specified.

Buffer Size (29-33)

Required entry if Record Format is D, V, J, K, E, R, 1, 2, or S.

Enter the size of the largest variable length record. This entry must be valid, however, it is not used by VISION:Transact. Valid entries are 1-32760 or nnnnK where nnnn is 1-9999 except for record formats F and I where nnnn is 1-31.

IT Statement - Item Tracking

Use the IT statement in a definition run or application generation run to specify an expiration date or retention period for items created or updated on the library in this run. IT statements are also used to specify a creator or updater of an item that is input in the run.

Run Name (1-8)

Optionally, you may enter the name of the run.

Statement Type (9-10)

Required entry

Enter IT.

Generic Item Name? (11)

Depending on whether the item name you are providing is generic or the full name, enter one of the following:

Entry	Result
Blank/N	The entry in Item Name is the full name.
Y	The entry in Item Name is generic.

Type Code (12-13)

Specify the type of item for which this IT statement is to be applied by entering one of the following codes:

Code	Description of Item Type
Blank	All items associated with the Item Name entry.
FD	All file definitions associated with the Item Name entry.
RQ	All requests associated with the Item Name entry.
RG	All request groups associated with the Item Name entry.
SD	All screen definitions associated with the Item Name entry.

Item Name (14-21)

Required entry

If Y is entered for Generic Item Name [11], enter a generic name (leading characters of the item name), not the full name. Otherwise, enter the specific or full item name.

Expiration Date/Retention Period (30-35)

Specify the date that the item expires or the number of days the item is to be retained in the library.

Entry	Results
nnnnn+	Enter one to five numerics, right-justified, followed by a plus sign to retain the item a specified number of days.
mmddyy	Specifies the date (TODAY flag format unless changed in M5PARM) you want the item to expire. where: mm=month dd=day 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)

Identify the person responsible for cataloging or updating an item with from 1 to 20 alphanumeric characters.

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.

Ln Statement - Comment

In a definition run or application generation run, use an Ln statement within a file definition to provide comments.

File Name (1-8)

Required entry

Enter the name of the file definition that you are documenting.

Statement Type (9-10)

Required entry

Enter Ln where n is a number from 1 to 9.

Segment/Field Name (11-18)

Required entry

Enter the name of a field from an L0 statement that this Ln statement is documenting.

Delete? (19)

To delete an existing comment enter Y; otherwise leave blank.

Comment (44-59)

Provide documentation for fields. Enter any helpful information.

LR Statement - Logical Relationship

The logical relationship statement is used to define the logical relationships between the different DB2 or SQL/DS tables used as segments in a logical record. It explicitly establishes the relationship between the rows of the different DB2 or SQL/DS tables. An LR statement must be provided for each non-root segment in a relational database.

File Name (1-8)

Required entry

Enter the same name as entered on the FD statement.

Statement Type (9-10)

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 relation conditions.

Delete? (19)

Allows you to add or delete an entire logical relationship definition.

Entry	Result
Blank/N	Definition is added to the library.
Y	Definition is deleted from the library. You must also enter the file name, segment name and sequence number.

Sequence Number (20-22)

Indicates the order of the statements in the definition. Duplicate sequence numbers within the same segment name are not permitted. Sequence numbers are required for segments containing multiple LR statements.

Entry	Result
Up to three alpha, numeric, or alpha-numeric characters	Sequence is in EBCDIC ascending order.

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 and is processed first 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 field in the DB2 or SQL/DS table to be compared to the item in positions [49-66] of this statement. This name is the external DB2 or SQL/DS column names as defined on the DB2 or SQL/DS catalog.

Operation (45-46)

Required entry

Specifies the logical operation to be used in comparing the external field named in positions [27-44] to the internal VISION:Transact field or value in positions [49-66] of this statement. The operator you specify results in a select (true) or non-select (false) condition. If the complete expression composed of all the LR statements for a segment is met (true), the DB2 or SQL/DS table row is selected and becomes a segment in the VISION:Transact record. If the condition is not met (false) the row is not selected.

Code	Conditions to be Met
EQ	External field equal to internal field or value.
NE	External field not equal to internal field or value.
LT	External field less than internal field or value.
GT	External field greater than internal field or value.
GE	External field greater than or equal to internal field or value.
LE	External field less than or equal to internal field or value.
NL	External field is null. Positions [49-66] must be blank.
NN	External field 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 field name or constant to be compared to the DB2 or SQL/DS table field named in positions [27-44]. This entry must correspond to the qualifier in position [48].

Qualifier	Entry
Blank	A field name that has been defined in a higher level segment. If the field named here maps a DB2 or SQL/DS table column, then the original contents of the field as retrieved from the table are used for the relational test in the SQL WHERE clause. If the field named here is an unmapped field (that is, the field does not name a DB2 or SQL/DS column), then the current contents of the field as modified by application logic are used for the relational test in the SQL WHERE clause. The use of unmapped fields on the LR statement allows the programmer to procedurally set the contents of the field prior to the retrieval operation for the segment (table) which uses the field. Unmapped fields, thus, allow tables to be related based on calculated or derived argument values.
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.

LS Statement - Segment Definition

In a definition run or application generation run, use an LS statement within a file definition to name a segment type for a database or pseudo file. For DB2 or SQL/DS logical records you must identify each table in the logical record as a segment on an LS statement.

File Name (1-8)

Required entry

Enter the name of the file definition as it appears on the FD statement.

Statement Type (9-10)

Required entry

Enter LS.

Segment/Field Name (11-18)

Required entry

Enter the name of the segment.

For an IMS or fast path database, enter the name of the segment as it appears on a SEGM statement of its DBD.

For all other files, enter the name you want to assign to the segment under VISION:Transact.

Delete? (19)

To delete a segment name from an existing file definition, enter Y. To modify a segment name from an existing file definition, enter Y and specify the new segment name on a subsequent LS statement.

Segment Number (20-21)

Number the segment from 1 to 99. If this entry is left blank, the default is 1. Together all the segment numbers in the file definition specify the hierarchy of the file. The numbers assigned to lower level segments must increase as you move from the top to bottom and then left to right in the file hierarchy. All dependent segments must have a segment number larger than the parent segment and smaller than any of the following segments.

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

Suppress Duplication? (30)

This entry is used for DB2 or SQL/DS logical records (Record Format S). It alters the way in which the logical records are constructed in a logical data view. When the relational table contains duplicate rows, they can be eliminated or included in the logical record.

Entry	Result
Blank/N	Duplicate rows are to be included as duplicate segment occurrences.
Y	Eliminate duplicate rows as segment occurrences.

Segment Order (31)

Enter one of the following codes to specify how to order the occurrences of the segment or to indicate a sequential dependent of a data entry database (DEDB).

Entry	Result
Blank	Specifies no segment ordering.
A	Specifies that segments be ordered in ascending order by their key field(s) within their parent.
D	Specifies that segments be ordered in descending order by their key field(s) within their parent.
S	Specifies for an IMS data entry database (DEDB) only that this segment is a sequential dependent. Only one sequential dependent is valid, and it must be in level two.

Creator ID or Authorization ID (36-43)

This entry is used for DB2 or SQL/DS logical records. It specifies the authorization ID of the person that created the relational table which is to be used to build the logical record.

If a fully qualified table or view name is to be used, enter the authorization ID of the person who created the table or view. If all relational tables in the logical record have the same creator, then this entry can be left blank and the authorization ID of the creator can be specified on the RF statement for the file during APPGEN.

If an unqualified table or view name or a synonym name is to be used, this entry must be left blank. In this case, VISION:Transact's generated SQL statements will use the unqualified name.

It is important to note that when an unqualified table, view, or synonym name is used, the VISION:Transact APPGEN process implicitly qualifies the name with the authorization ID of the person who submitted the APPGEN run. The column names in the file definition are validated by VISION:Transact based on the synonym, table, or view name found for the user doing the APPGEN. When the BIND is performed for the application plan, DB2 or SQL/DS implicitly qualifies an unqualified name with the authorization ID of the person who submitted the BIND run. Thus, DB2 or SQL/DS validates the VISION:Transact application's generated SQL statements using the synonym, table, or view name found for the user doing the BIND, whereas VISION:Transact's validation is performed using the synonym, table, or view name found for the user doing the APPGEN. If the authorization ID of the APPGEN job is different from the BIND job, different synonym, table, and view names would be used by VISION:Transact and SQL. This can cause problems if the column definitions for each are not the same.

DB2/SQL External Name (44-61)

Required entry for DB2 or SQL/DS logical records

Enter the name of the DB2 or SQL/DS object from which this segment is mapped. This entry can be the name of a table, view, or synonym.

L0 Statement - Field Definition

In a definition run or application generation run, use an L0 statement within a file definition to specify and name a field for a database, pseudo file, working storage area, or PCB mask. You can also identify the external field of a relational database to be retrieved and converted to a logical record.

File Name (1-8)

Required entry

Enter the name of the file definition as it appears on the FD statement.

Statement Type (9-10)

Required entry

Enter L0.

Segment/Field Name (11-18)

Required entry

Enter the name of the field. For an IMS or fast path database, search or key fields names must match those in the DBD. If the search or key field named here does not match the DBD field name, the ALIAS entry must be used to supply the field name from the DBD.

Delete? (19)

To delete a field from an existing file definition, enter Y. To modify a field from an existing file definition, enter Y and specify the new field on a subsequent L0 statement. In an application generation run, file definitions are not updated on the library, they are only temporarily modified or created.

Segment Number (20-21)

Enter a number from 1 to 99 to identify the location of the segment in the hierarchy of the file. If this entry is left blank, the default is 1. The numbers assigned to lower level segments must increase as you move from the top to the bottom and then left to right in the file hierarchy. All dependent segments must have a segment number larger than parent segment and smaller than any of the following segments.

Define only a root segment (segment number 1) for a working storage or a PCB mask file definition.

For an IMS main storage database (MSDB fast path database), define only a root segment. For an IMS data entry database (DEDB), define a root segment and possibly a sequentially dependent second level segment, and any direct dependents.

Level Number (22)

Enter a number from 1 to 9 to define the subordination of the segment in the hierarchy of the file. The default is level number one, and there can only be one segment at level one, the root. Each field in a segment must be specified with the same level number.

For a working storage area, a PCB mask, or a main storage database (MSDB), only level one (the root) is valid.

Field Location (23-26)

Enter a number from 1 to 9999 to indicate the starting location of the field within its segment. This number is the byte number of the leading or high order byte in the field where 1 is the first byte of the segment. The starting location of a field is relative to the beginning of its segment. Leave this entry blank if you are defining a virtual key field.

The record key of an ISAM file cannot begin in byte 1.

Field Length (27-29)

Required entry

From the following ranges, enter the number of bytes occupied by the field:

Entry	Length (in bytes)	Representation
Blank/C	1-255	1 to 255 characters
Z	1-15	1 to 15 digits
P	1- 8	1 to 15 digits (plus sign)
E	4 only	Range 10^{-78} to 10^{75}
F	1-4	1 byte ranges from -128 to 127 2 bytes range from -32,768 to 32,767 3 bytes range from -8,388,608 to 8,388,607 4 bytes range from -2,147,483,648 to 2,147,483,647

For field type P, enter the number of bytes, not digits. Fixed point numbers (type F) are carried in two's complement notation. Virtual key fields for IMS databases must specify the same length as the XDFLD specification in the DBD (combined length of all fields specified by the SRCH parameter of the XDFLD statement in the DBD).

For CICS TS, a virtual key for VSAM ESDS and RRDS files must be defined as 4 bytes fixed point binary (field type F).

Field Type (30)

Enter one of the following alphabetic codes:

Entry	Resulting Field Type
Blank/C	Character string.
Z	Number in zoned decimal representation.
P	Number in packed decimal representation.
E	Floating point.
F	Fixed point number in two's complement binary format.

When a record or segment is inserted in a database, the application initializes character fields to blank and numeric fields to zeros.

Data types for DL/I and SQL along with their equivalent VISION:Transact field types are shown in [Figure 2-1](#).

DL/I Data Type	Data Type Description	VISION:Transact Type	
C	Character string	C or Z	
P	Packed decimal	P	
X	Hexadecimal	F or E	

SQL Data Type	Data Type Description	VISION:Transact Type	Length
Smallint	2 byte signed binary	F	2
Integer	4 byte signed binary	F	4
Float	8 byte floating point	E ¹	8
Decimal (n,d)	Signed packed decimal n = digits, maximum 15 d = decimals	P ²	1-8
Char (n)	Character data (1-254 bytes)	C	1-254
Varchar (n) long varchar	Variable length character (1-32674 bytes)	C ³	1-255
Date, time timestamp	Date and time type fields Date Time Timestamp	C C C	10 8 8
Graphic types	Double-byte characters	C	26
		Not supported	

Note:

1. Only 4-byte floating point allowed (single precision).
2. Only 9 decimals allowed.
3. Character fields may only be 255 bytes.

Figure 2-1 DL/I and SQL Data Types and Their VISION:Transact Equivalents

Segment Key (31)

Enter one of the following:

Entry	Result
Blank	For a non-key and non-search field.
1 to 9	Every segment must have at least one key field. To determine the order, major to minor (1 to 9) of the key field in the segment. For an ISAM or KSDS VSAM file (type I, J or K) or DL/I database only one key field is valid for the root segment. The key of an ISAM record cannot begin in byte 1. Key fields in DB2 or SQL/DS logical records must specify the DB2/SQL EXTERNAL NAME in positions [44-61].
S	To define a field in a DL/I database as a search field. This specification must match a search field specification in the DBD. This entry applies only to DL/I databases.
V	For a virtual key field in the root segment only. This can be used to access a DL/I database via a secondary index. A virtual key must be used to define a terminal-related MSDB. For CICS TS, VSAM ESDS and RRDS files must be defined with a 4-byte binary (field type F) virtual key. In these cases, no other key fields can be defined in the root segment. Refer to Chapter 10 of the <i>Advantage VISION:Transact Reference Guide</i> for a detailed explanation of the use of virtual key fields. This entry is applicable only to DL/I or MSDB databases or VSAM ESDS or RRDS files.

Decimal Places (33)

Enter the implied number of digits to the right of the decimal place according to the following rules:

Entry	Rules
C	Leave entry blank, decimal places are invalid.
Z	Decimal places must not exceed the field length.
P	Decimal places must not exceed twice the field length minus one.
E	Leave entry blank, decimal places are invalid.
F	Decimal places must not exceed twice the field length plus one.

A zero-valued field specified with decimal places is displayed as a decimal point followed by a zero for each decimal place.

Count Field For Segment (34-35)

For a field used to count variable occurrences, enter the segment number (right-justified) of the variably occurring segment. If this field is not a count field, leave the entry blank. This field applies to type P, Z, or F fields only.

A file with variable length records is implemented with segments having a variable number of occurrences. These occurrences are counted in a field, referred to as a count field, in the parent of the variably occurring segment. Each time an occurrence is inserted or deleted in a variably occurring segment, VISION:Transact automatically updates its designated count field in the parent segment. Your application must not update such a count field, but it may reference it for comparisons.

Count fields are often used in VSAM databases and can be used for variable pseudo file definitions, but they cannot be used in IMS, fast path, or relational databases, or files with fixed length records.

Never over-define a count field as a character or part of a character field; instead, over-define a character field as a count field.

This Segment Occurs N Times (36-38)

In the key field of a fixed occurring segment, enter the number of times (right-justified) this segment occurs within each occurrence of its parent. In case of multiple key fields, this entry must be used in the first key field defined for the segment and it may be used, unchanged, in all key fields defined for the segment.

This entry specifies the fixed number of times a segment occurs for each occurrence of its parent. If a segment key field is blank (zero if numeric), the segment is considered non-existent and cannot be accessed.

Leave this entry blank in a file definition for an IMS, relational, or fast path database, working storage, or PCB mask.

Output Edit/Length (42-43)

Leave this entry blank, and VISION:Transact automatically computes an output edit length large enough to display the maximum number of positions needed by this field. Otherwise, enter the number of screen positions you need to display the value of this field. Add one position if a leading plus or minus sign is possible, and another position of a decimal point is possible.

This entry specifies the number of characters used to display this field. It is better to leave this entry blank and specify underscores on the screen layout to restrict the size of the value displayed for the field. Any entry here has no effect on the size of the screen field used to display the value of this field.

Alias (44-51)

This specification is used when defining an IMS or fast path database with non-unique segment key/search field names or segment key/search field names that contain special characters.

Enter the name (1-8 bytes) which corresponds to the name specified in the DBD.

This entry, used on an L0 statement which defines a segment or virtual key/search field, provides the name specified in the DBD. VISION:Transact uses this name only to construct segment search arguments (SSAs). The VISION:Transact source code must reference this field by the name entered in positions [11-18].

DB2/SQL External Name (44-63)

This entry is used to identify the DB2 or SQL/DS column name in the relational table that is to be mapped into the logical record field named in positions [11-18] of this statement. This entry is required for all fields that are defined as segment keys. Enter the 1-18 character SQL column name. Although not required unless this definition is used by other Computer Associates products, an SQL delimited column name of up to 20 characters may be entered. An SQL delimited column name is a column name enclosed in double quotation marks, as it contains special characters or is an SQL reserved word.

If this entry is left blank, the field will exist in the VISION:Transact logical record only, no transfer of data takes place between the relational table and the VISION:Transact logical record buffer. This type of field is referred to as an unmapped field. Unmapped fields can be used on LR statements which allows programmers to relate tables based on calculated or derived argument values, effectively creating a dynamic rule.

PR Statement - Processing And Selection

In a definition run or application generation run, use a PR statement in a request to provide procedural code to manipulate application data. PR statements can also be used to call programs that are not generated by VISION:Transact.

In a definition run, PR statements can be cataloged in a request or request group.

In an application generation run, PR statements can be cataloged in a request (not in a request group), retrieved as part of a cataloged request or request group, or they can be input to requests which are for temporary use in the APPGEN.

To code a CALL statement skip to [Call, PLI or DLI \[15-18\] on page 2-44](#).

OPERATOR	LOGIC LEVEL	CONN	OPER A Q,FIELD	OPER B Q,FIELD	OPER C Q,FIELD	PARTIAL FIELD
Relational	Y	Y/R	R,R	R,R	N,N	Y
Branching	N	N	N,N	N,R	N,N	N
Arithmetic	N	N	R,R	R,R	R,R	Y
Replacement	N	N	Y,Y	Y,Y	R,R	Y
Messages (SE) (MS)	N N	N N	R,R Y,Y	N,N R,R	N,N N,N	N Y
Trace	N	N	N,N	R,R	N,N	N
Y = This entry may be specified. R = This entry is required. N = This entry is prohibited. Y/R = This entry may be specified or required based on the value of the CONREQ parameter in M5PARM.						

Figure 2-2 Summary of PR Statement Entries

Request Name (1-8)

Required entry

Enter the same request name used in the ER statement of this request.

Statement Type (9-10)

Enter PR.

Statement Label (11-13)

If this statement is the destination of any branch in this request, enter a 1-3 position alphanumeric label. The label must not equal END, since END is a reserved branch location at the end of the request. If this entry is blank, then this PR statement cannot be the destination of a branch operation.

Logic Level (14)

Specifies the sequence of processing multiple and/or conditions in a relational expression. If this PR statement has an O (OR) or A (AND) logic connector in position [15], enter its logic level (from 0 to 9). The logic level number is similar to using parentheses in an algebraic expression. If you leave this entry blank, a logic level of 0 is assumed. If no logic connector is used or assumed on this PR statement you must leave this entry blank. To determine the logic level, follow these steps:

1. Write the conditions of the logical expression using ANDs, ORs, and parentheses, where appropriate. This expression may eventually be coded using multiple PR statements.
2. Count the pairs of parentheses surrounding the connector associated with this PR statement. Enter this count as the Logic Level [14] of this PR statement.

If logic levels of several related logic connectors are the same or are not specified, AND takes precedence over OR. For example, A OR B AND C is interpreted as A OR (B AND C).

Connector (15)

Enter A for AND or O (alpha O) for OR to join the relational operation (EQ, NE, GT, LT, LE, or GE) on this PR statement with the one on the previous PR statement. If the Connector entry is blank and CONREQ=N is specified in M5PARM, the default OR connector is assumed. The Connector entry cannot be blank when joining two or more relational statements if CONREQ=Y is specified in M5PARM. See the *Advantage VISION:Transact for CICS Installation Guide* for more information on M5PARM.

This entry must be blank when this PR statement has no logical operation or the previous PR statement has no logical operation.

The AND logical connector has precedence over the OR connector. Use the AND connector to join coincident logical expressions, and use the OR connector to join alternative logical expressions.

Operand A (16-24)

Enter the qualifier and the name of a field used for comparison or processing.
Leave blank for branching and trace operators.

Qualifier A (16)

Enter one of the following qualifiers:

Qualifier	Source of Field/Literal
Blank/N	Database or pseudo file whose File Type is FILE.
1-9	Database or pseudo file whose File Type is FILE1 to FILE9.
S	Input screen field (invalid for request type O).
T	Temporary field.
F	Flag field (VISION:Transact system field).
W	Working storage area.
V	PCB mask field.
C	Use with the literal NOROLLB in [17-23] for the MS operator only.

If the qualifier refers to a database or pseudo file field, refer to the RF statement which specifies its accessibility and determines its synchronization method. If it is screen synchronized, fields are available to type 1 requests. If it is dynamically synchronized, its synchronizing field must be available to this request type. See the *Advantage VISION:Transact Reference Guide* for more details.

If database updates are not to be backed out for this execution of the MS operator, enter a C in Qualifier A and the literal NOROLLB in Field Name A. Note that VISION:Transact performs physical updates after type M requests. Therefore, an MS operator which may or may not back out updates only affects VISION:Transact files when issued during output processing.

Field Name A (17-24)

Enter the name of the field that is used for comparison or processing. Enter the field name from the file or screen definition glossary, temporary field, or flag field.

If database updates are not to be backed out for this execution of the MS operator, enter NOROLLB in [17-23].

Operation (25-26)

Required entry

There are six types of operations: relational, arithmetic, replacement, branching, error message, or trace. Single character operators may appear in position [25] or [26].

Relational Operators: To compare two operands (Operand A and B) enter a relational operator to be used in a relational expression which results in a TRUE or FALSE condition. The relational condition is evaluated by a conditional branch operation in a subsequent PR statement. If no conditional branch operator and no logical connector follows, an NS END conditional branch operator is assumed. You can connect relational operations using AND and OR connectors and from zero to nine logic levels.

Branch Operators: To alter the normal sequence of operation from one PR statement to the next, enter a branching operator in positions [25-26]. When a branch is taken, Operand B specifies the destination location. A PR statement with a branch operator must have Operand A blank.

Arithmetic Operators: An arithmetic operation requires the specification of all three Operands. The arithmetic operation is performed on the fields specified in Operands A and B and places the result in the field specified in Operand C.

Replacement Operator: In replacement operations, the value of the field named by Operand C is updated with the value of the field named by Operand A or Operand B, exclusively. The replacement operator is R for “replace” and it can be entered in position [25] or [26].

Message Operators: The Set Error (SE) and Message (MS) operators are used to identify screen input fields having errors, and to re-display the input screen with an error message. Database updates are backed out unless an entry is made in Operand A of the MS operator.

Trace Operators: The trace operators are used to display literals or fields during online application execution. The literal or field can be displayed in character format (?C operator) or both character and hexadecimal format (?X operator) within the trace output. Note that trace operators are always valid within a request; however, they are only executed when procedure tracing is enabled via the ?T statement or contents of flag TPROC.

Enter one of the following operators:

Operator	Description
Relational Operators	
EQ	Operand A equals Operand B or matches the pattern in Operand B.
NE	Operand A does not equal Operand B, or does not match the pattern in Operand B.
LT	Operand A is less than Operand B.
GT	Operand A is greater than Operand B.
LE	Operand A is less than or equal to Operand B.
GE	Operand A is greater than or equal to Operand B.
Branch Operators	
GO	Unconditionally branch.
NS	Branch if the preceding logical condition is FALSE.
GS	Branch if the preceding logical condition is TRUE.
Arithmetic Operators	
+	Add Operand A to Operand B; put the sum in Operand C.
-	Subtract Operand B from Operand A and put the difference in Operand C.
*	Multiply Operand A by Operand B and put the product in Operand C.
/	Divide Operand A by Operand B and put the quotient in Operand C.
Replacement Operators	
R	Replace the value of Operand C with the value of Operand A or Operand B, exclusively.
Message Operators	
MS	Terminate the interaction and re-display the input screen with the value of Operand B displayed on the screen's message line. Database updates are backed out unless an entry is made in Operand A.

Operator	Description
SE	Increment the count in the ERROR flag, and specify an input screen field (Operand A) that is to be processed with error handling. Error handling can specify that the field be highlighted. Leave Operands B and C blank and do not use any partial fielding specifications. The SE operator cannot be used in a type O request nor in a subroutine called from a type O request.
Trace Operators	
?C	Display the contents of Operand B in character format within the trace output when procedure tracing is enabled.
?X	Display the contents of Operand B in both character and hexadecimal format within the trace output when procedure tracing is enabled.

Operand B (27-58)

Operand B can take on different roles depending on the operator in positions [25-26].

Operator	Role of Operand B
Relational	Compared with Operand A.
Branching	Provide branch destination.
Arithmetic	Added to Operand A, subtracted from Operand A, multiplied by Operand A, or divided into Operand A.
Replacement	Provide value to be placed in Operand C if Operand A is not used.
Message	If the MS operator is used, provide up to 79 bytes of the message text to be displayed, or if the SE operator is used, leave blank.
Trace	Provide a constant or field to be displayed in the trace output during application execution.

Operand B Qualifier (27)

Enter one of the following codes:

Entry	Result
Blank/N	Field from database or pseudo file whose File Type is FILE.
1-9	Field from database or pseudo file whose File Type is FILE1-FILE9.
S	Field from the associated input screen (invalid for a type O request).
T	Temporary field.
C	Constant character string field.
D	Decimal constant field.
W	Working storage area field.
V	PCB mask field.
P	Pattern (only valid for EQ or NE operators).

The trace operators do not permit the creation of implicit field definitions. Therefore, when a temporary field is used as Operand B of a trace operator, the temporary field must either be defined explicitly on an TF statement or implicitly on a preceding non-trace operator statement.

For a request to reference a field from a database that is dynamically synchronized, its synchronizing field must be accessible (see the *Advantage VISION:Transact Reference Guide*). Fields from databases that are screen synchronized are not available in type 1 requests.

Field Name B or Constant (28-58)

Enter the name of the field or the constant.

If the PR statement used a branch operator (GO, GS, or NS), specify one of the following branch-to locations (Operand B Qualifier must be blank):

Statement Label: Branch to a statement later in the same request. Operand B must specify the entire label (3 characters), including leading/trailing blanks/zeros. The label must appear in a subsequent PR statement of the same request.

END: Branch to the end of this request and terminate the processing of the current segment occurrence of the intermediate structure or database. In a type O request this bypasses data movement to output screen fields.

NEXT REQUEST: Branch to the end of the request and terminate all outstanding loops. In a type O request, terminate output processing for the current output screen. This branch is invalid in a type S request.

OUTPUT: Branch to output screen processing. This branch is valid only in a type O request.

SUB request-name: Branch to the subroutine request named request-name. Upon returning from the named request, begin with the statement following this branch statement.

RETURN: This branch is only used in a type S request to return to the request that invoked it.

BACK: Branch to the top of the current request. This branch is used for procedural looping or back-branching and relies on the Back-Branch Limit entry of the ER statement.

NEXT FUNCTION: Stop processing this request, skip other requests of this type, and branch to the next VISION:Transact function. This branch can be used in any request type (1, 2, M, O, or S). A type O next function branch stops building the output screen and outputs the partial output buffer. A type 1 next function branch skips to automatic input data validation. A type 2 next function branch skips to in-memory updates. A type M next function branch skips to update the databases.

SCREEN CONTROL: Stop processing this request, skip all other input requests, and input VISION:Transact functions (no database updates), set the GOSC flag to this request's type, and go to screen control. This branch is valid in any request type (1, 2, M, O, or S). A type 1 screen control branch skips screen synchronization and causes screen synchronized fields to be missing on the output screen; field actions are not performed. A type 2 screen control branch skips field actions but not segment actions. A type M screen control branch skips physical database updates, but not segment or field actions performed in storage. A type O screen control branch is treated as NEXT FUNCTION.

If the PR statement used an EQ or NE operator, you may use one of the following patterns for validation (a P must be used in Operand B qualifier):

Date Patterns:

.MMDDYY

.MMYYDD

.DDMMYY

.DDYYMM

.YYMMDD

.YYDDMM

.TODAY

.JULIAN

Numeric Patterns:

.NUMERIC

.INTEGER

.FLOAT

Refer to [Chapter 3, “Advantage VISION:Transact Flags and Patterns”](#) for more information on patterns.

Operand C (59-67)

Operand C can take on different roles depending on the operator in positions [25-26].

Operator	Role of Operand B
Relational	Operand C must be blank.
Branching	Operand C must be blank.
Arithmetic	The sum, difference, product, or quotient of Operands A and B are placed in Operand C.
Replacement	The value of Operand A or B is copied to Operand C.
Message	Operand C must be blank.
Trace	Operand C must be blank.

Operand C Qualifier (59)

Enter one of the following codes:

Entry	Result
Blank/N	Field from database or pseudo file whose File Type is FILE.
1-9	Field from database or pseudo file whose File Type is FILE1-FILE9.
S	Field from the associated input screen (invalid for a type O request).
T	Temporary field.
F	Flag field.
W	Working storage area field.

For a request to reference a field from a database that is dynamically synchronized, its synchronizing field must be accessible (see the *Advantage VISION:Transact Reference Guide*). Fields from databases that are screen synchronized are not available in type 1 requests.

Operand C Field Name (60-67)

Enter the name of the result field.

Partial Field (68-72)

Use these entries to specify processing for only part of a character field (this includes screen fields). Refer to [Figure 2-2 on page 2-32](#) to determine whether an operator allows partial field entries.

Starting Character (68-69)

Enter one of the following numbers or codes to specify where the partial fielding is to begin.

Entry	Partial Field Begins
Blank	No partial fielding.
1-99	Position of character beginning the partial field (must not exceed the length of the field).
LS	Current value of LSTART flag.
LN	Current value of LNUMBER flag.

Entry	Partial Field Begins
MS	Current value of MSTART flag.
MN	Current value of MNUMBER flag.
PS	Current value of PSTART flag.
PN	Current value of PNUMBER flag.
RS	Current value of RSTART flag.
RN	Current value of RNUMBER flag.

Number of Characters (70-71)

Enter the number of characters forming the partial field in one of the following ways:

Entry	Result
Blank	No partial fielding.
1-99	Number of characters forming the partial field (up to the maximum length of the field).
LS	Current value of LSTART flag.
LN	Current value of LNUMBER flag.
MS	Current value of MSTART flag.
MN	Current value of MNUMBER flag.
PS	Current value of PSTART flag.
PN	Current value of PNUMBER flag.
RS	Current value of RSTART flag.
RN	Current value of RNUMBER flag.

Operand A, B or C (72)

Identify the operand to which the partial fielding applies.

Entry	Result
Blank/A	Operand A.
B	Operand B.
C	Operand C.

Note: The Operand specified must be one that is valid for the operation (e.g., C is not valid for a relational operator).

PR Statement - Call Option

To procedurally call an external subroutine from a VISION:Transact generated application, code a Call PR statement in a request. You can issue calls to user written routines, to the DL/I database system modules, or to subroutines written in PL/I. Just supply the appropriate keyword CALL, DLI, or PLI. If applicable, name the called routine and provide a list of its parameters.

Request Name (1-8)

Required entry

Enter the same request name used in the ER statement of this request.

Statement Type (9-10)

Enter PR.

Statement Label (11-13)

If this statement is the destination of any branch in this request, enter a 1-3 position alphanumeric label. The label must not equal END, since END is a reserved branch location at the end of the request. If this entry is blank, then this PR statement cannot be the destination of a branch operation.

Logic Level (14)

Required entry

Enter the system delimiter (default is #).

Call, PLI or DLI (15-18)

Required entry

Enter one of the following keywords followed by a blank:

Entry	Result
CALL	Calls a non-VISION:Transact generated subroutine.
DLI	Calls DL/I using a PCB mask.
PLI	Calls a PL/I subroutine under IMS TM only.

Load Module or CSECT Name (20-27)

Required entry

For the CALL or PLI keyword, enter the load module or CSECT name, followed by one or more blanks, which receives control when the call is executed. The module must be available when the VISION:Transact generated application executes. In some instances the called routine may be linked into the generated application (refer to the LKEDCALL specification on the RC or RP statement).

For the DLI keyword, this entry is invalid and assumed to be ASMTDLI.

Parameters

Enter any parameters (fields) to be passed to the invoked module or routine, using the following format:

Qualifier#field name

Observe the following rules:

1. You may specify any number of parameters (fields) on multiple PR statements, as long as one of them is on the first CALL statement and no parameter is split among two statements.
2. For a parameter, use any qualifier that is valid with Operand B of the PR statement. If you specify no qualifier, the parameter is assumed to be a field from the database whose File Type is FILE.

3. Precede each parameter by at least one blank and separate its qualifier from its field name with the system delimiter.
4. If the parameter is a character string constant (having a C qualifier), avoid embedded or trailing blanks.

Note:

IMS TM: All types of calls (CALL, DLI, and PLI keywords) must have parameters.

CICS: A call to IMS (DLI keyword) must have parameters. A call to a non-VISION:Transact generated subroutine (CALL keyword) may or may not have parameters, as described below.

CICS: There are two methods available for passing data to a called program which is not link edited with the VISION:Transact application.

1. If data is to be passed through the COMMAREA, code two parameters on the CALL statement. The first parameter names the field that defines the first byte of the COMMAREA. The second parameter defines a 2-byte fixed point binary field that contains the length of the COMMAREA to be passed to the called program.
2. If data is to be passed through the CICS TWA, then no parameters are coded on the CALL statement. In this case, data is placed in VISION:Transact working storage fields as these reside in the CICS TWA.

If the called program is link edited with the VISION:Transact CICS application, the program must be written in assembler language and contain no CICS macros or commands. The desired parameters are specified on the call statement using the syntax as previously described.

See Chapter 7 of the *Advantage VISION:Transact for CICS Environment Guide* for more information on calling user written programs.

Continuation (72)

To continue the call statement to another PR statement, enter a non-blank character. Code the system delimiter in position [14] of the next PR statement and continue in position [15].

RC Statement - Run Control

The run control (RC) statement must be the first statement input in a definition run, application generation run, or source statement retrieval run. The RC statement must not be used in a library utility run.

In a definition run used to catalog screens, the RC statement specifies the target monitor. This information is used to determine whether MFS or BMS source statements are to be output for the screens.

In an application generation run, the RC statement specifies the target monitor, names the application, the transaction, and the PSB, when applicable. It also specifies special processing requirements for the application.

In a source statement retrieval run, the RC statement can request that a list of the retrieved source be printed.

Run Name/Transaction Code (1-8)

Required entry

In a definition run or source statement retrieval run, enter any convenient name.

In an application generation run, the role of this entry depends on the target monitor.

IMS TM: Positions [1-8] specify the IMS TM transaction code. It is also used for the load module name and the PSB name, unless its overridden by the PSB Name entry, positions [41-48] of this statement. It must be the same name as used on the SD statement (Input Transaction Code) or in the TRANCODE flag for each of the input screens used in the application.

CICS TS: Positions [1-4] of this entry specify the CICS transaction code. Positions [1-8] are also used for the load module name and the PSB name unless overridden by the Program Name, in positions [49-56], or the PSB Name, in positions [41-48] of this statement.

Statement Type (9-10)

Required entry

Enter RC.

Library Open (28)

Use this entry to indicate how the primary VISION:Transact library should be opened during an APPGEN run.

Enter one of the following codes:

Entry	Result
Blank	Use the default value specified by the LIBOPNA keyword in M5PARM.
U	Open the library for update, allowing screen definitions and individual requests to be cataloged during the APPGEN run.
R	Open the library for read only.

System Delimiter (29)

In a definition run or an application generation run, enter any character to override the system delimiter named in M5PARM. Leave this entry blank and the default system delimiter specified in M5PARM is assumed.

Debug Level (30)

In an application generation run only, use this entry to control the way an application targeted for IMS TM handles an unexpected condition, such as an unexpected status code from IMS.

CICS: This entry is ignored.

IMS TM: Enter one of the following codes:

Entry	Result
0	Dump and ABEND. The application sends an error screen to the error destination terminal to describe the unexpected condition.
Blank/1	Terminate normally. The application sends an error screen to the originating terminal and the error destination terminal to describe the unexpected condition. The application may abend if it cannot recover from the unexpected condition.

The information on the error screen depends on the unexpected condition. It may describe an unexpected DL/I return code, an unexpected interrupt or an internal VISION:Transact error.

Source Statement Listing (31)

Enter one of the following codes:

Entry	Result
Blank/N	Format and list all source statements.
C	Format and list all source statements and produce screen definition glossaries for all screens used in the APPGEN (APPGEN run only).
G	Format and list source statements, produce screen glossaries, and additional internal information for the purpose of submitting problem resolution documentation to Computer Associates.
X	Produce card-image (unformatted) listing of all statements output to M5SSOUT (source statement retrieval run only).

LKEDCALL Programs (32)

Use this entry to indicate which called programs are to be link edited with a VISION:Transact application.

Enter one of the following codes:

Entry	Result
Blank	Use the default value specified by the LKEDCIS or LKEDIMS keyword in M5PARM. The keyword used depends on the target monitor for this application. See the <i>Advantage VISION:Transact for CICS Installation Guide</i> for information on M5PARM.
N	Only those called programs named on RP LKEDCALL statements are to be link edited with the VISION:Transact application.
Y	All called programs are to be link edited with the VISION:Transact application.

System Rollback Control (33)

This entry controls database backout (rollback processing) when errors are detected by VISION:Transact and the error cycle is invoked. VISION:Transact invokes the error cycle upon encountering one of the following errors:

1. An input screen field fails validation specified in the screen definition (e.g., input required, numeric, or user validation fails).
2. An input screen field cannot update its destination (e.g., the result of a field action in the screen definition would cause the destination to become invalid).
3. An error occurs during retrieval or update of a database segment (e.g., the segment occurrence is not on the database or the segment cannot be inserted because it already exists).

This entry only controls rollback processing if any of the above errors are detected by VISION:Transact. To control rollback processing for errors detected by the programmer during request processing, Operand A of the MS operator must be used. To control rollback processing for errors detected by the programmer during procedure processing, the NOROLLBACK option of the MSG command must be used.

Enter one of the following codes:

Entry	Result
Blank/A	Database updates are always backed out. This option is recommended to preserve the integrity of databases processed by VISION:Transact.
M	Database updates are backed out when an error is detected after type M request or procedure processing. This type of error can occur during VISION:Transact's physical update, screen control, or output processing. This option preserves database integrity only for those databases processed by VISION:Transact. It is the programmer's responsibility to ensure that any database updates performed external to VISION:Transact during type 1, 2, and M processing are backed out if required. Updates that are performed external to VISION:Transact are those done by user DL/I calls or called user written external subroutines.
N	Database updates are not backed out. This option does not preserve the integrity of databases processed by VISION:Transact or processed external to VISION:Transact (i.e., DL/I calls or called user subroutines).

CICSKEY Handling (34)

CICS:

This entry indicates how the Clear key and PA keys are to be handled during application execution under CICS. This entry overrides the default CICSKEY specification in M5PARM. Refer to the *Advantage VISION:Transact for CICS Installation Guide* for detailed information on the CICSKEY specification in M5PARM.

Enter one of the following codes:

Entry	Result
Blank	The default value for the CICSKEY keyword in M5PARM is used.
N	Clear and PA keys are intercepted as a request for application termination.
Y	Clear and PA keys are not intercepted. This option causes the VISION:Transact application logic to be executed when the Clear or PA keys are pressed, just as is done for the Enter and PF keys.
C	The Clear key is not intercepted as a request for application termination. This option causes the VISION:Transact application logic to be executed when the Clear key is pressed. With this option, PA keys are intercepted and treated as a request for application termination.
P	PA keys are not intercepted as a request for application termination. This option causes the VISION:Transact application logic to be executed when a PA key is pressed. With this option, the Clear key is intercepted and treated as a request for application termination.

When a Y, C, or P entry is specified, the application code is executed as is done for the Enter and PF keys. Since CICS does not send input screen data to an application program when the Clear or PA keys are pressed, application programs must have appropriate logic to handle processing of these keys. For more information on the subject, see the *Advantage VISION:Transact for CICS Environment Guide*.

Message Processing (36)

IMS TM: Use this entry in an application generation run to specify how messages are to be processed under IMS TM. If there is no default for this entry provided in M5PARM, this is a required entry. To use the default specified in M5PARM leave this entry blank. See the *Advantage VISION:Transact for CICS Installation Guide* for more information on M5PARM.

IMS TM: Enter S to specify this as a single message processing application. The input message is read and processed, and the application terminates.

Enter M to specify this as a multiple message processing application. After the input message is processed, the application reads the message queue. The application terminates when there are no more messages on the queue or when the PROCLIM on the TRANSACT macro is met.

CICS: This entry is ignored in CICS TS.

Positional PCB? (37)

This entry specifies the type of PCB selection used by the generated application.

Enter one of the following codes:

Entry	Result
Blank/N	PCB is selected for a database by its DBD name (file identifier entry on the FD statement) and the database processing options.
Y	PCB is determined by the relative position of the file's RF statement with respect to the PSB. See the <i>Advantage VISION:Transact for CICS Environment Guide</i> for more details.

Run Type (38)

This entry specifies whether this is a definition run, application generation run, or source statement retrieval run.

Enter one of the following codes:

Entry	Result
Blank/D	Definition run or catalog maintenance run.
G	Application generation run.
S	Source statement retrieval run.

Target Monitor (39)

In a definition run that catalogs screens or in an application generation run, this entry specifies whether the input code is targeted for IMS TM or CICS TS.

Enter one of the following codes:

Entry	Result
C	CICS TS is the target monitor. (COMMAREA program or terminal program.)
I	IMS TM is the target monitor.
Blank	Use the default monitor coded for the TPMON keyword in M5PARM.

Target Operating System (40)

This entry is only used in an application generation run to specify the operating system that the application is to run on. If the target monitor is IMS TM, this entry is ignored since only z/OS is possible.

Enter one of the following codes:

Entry	Result
O	Target operating system is z/OS.
Blank	Target operating system is determined by an entry in M5PARM.

PSB Name (41-48)

Use this entry only in an application generation run, to specify a PSB or (under IMS TM only) the program/load module name for the application that differs from the transaction code specified in positions [1-8] of the RC statement.

IMS TM: Enter the name (left-justified) of the PSB and the program/load module for this application. Leave this entry blank and positions [1-8] of the RC statement name the PSB and program/load module of this application.

CICS: Enter the name (left-justified) of the PSB for this application. Leave this entry blank and positions [1-8] of the RC statement name the PSB of this application.

CICS TS: Enter the name (left-justified) of the PSB for this application in positions [41-47] (seven characters or less). Leave this entry blank and positions [1-7] of the RC statement name the PSB of this application.

Program Name (49-56)

Use this entry only in an application generation run, to specify the program/load module name for this CICS application that differs from the transaction code specified in positions [1-8] of the RC statement.

IMS TM: This entry is ignored.

CICS: Enter the name (left-justified) of the program/load module for this application. Leave this entry blank and positions [1-8] of the RC statement name the program/load module of this application.

RF Statement - Application Files

Used in an application generation run only, an RF statement specifies the way an application file is processed. It also associates a qualifier and a file definition with the file, so that application statements can reference fields in the file.

The variety of files possible in an application include databases, pseudo files, working storage areas, and PCB masks.

When processing a pseudo file observe the following rules:

- A pseudo file is treated like any other screen synchronized database except no physical I/O occurs. Also, pseudo files are available to type 1 requests or procedures; all fixed occurring segments are available to type 1 requests or procedures.
- Any file definition with a Record Format of F or V may be processed as a pseudo file.

When processing a working storage area or a PCB mask observe the following rules:

- The last byte, according to the record length specification, must be defined to VISION:Transact in an L0 statement.
- The file definition must define a one-level data structure.
- When two or more working storage areas or PCB masks are used by an application, all fields with the same qualifier (W for working storage, V for PCB mask) should be uniquely named.

[Figure 2-3](#) summarizes the RF statement codes used to identify the type of synchronization used to process databases and pseudo files. If a database is defined as both screen and dynamically synchronized, combine the appropriate entries. Note that a pseudo file cannot be defined as a DSDB; no sync fields or qualifier entries are allowed.

FILE TYPE FILEn (n=blank or 1-9)	SYNC FIELD 1	SYNC FIELD 2	SYNC QUALIFIER	SCREEN SYNC MATCH/ UPDATE
SSDB ¹ read only				M
SSDB ¹ updatable				U
DSDB ² direct read	name		note ³	
DSDB ² range browse	name ¹	name ²	note ³	
DSDB ² generic browse	name ¹	name ¹	note ³	
Pseudo file				P

1. SSDB = Screen Synchronization Data Base.
2. DSDB = Dynamically Synchronized Data Base.
3. See Sync Qualifier specification for allowable entries.

Figure 2-3 Summary of RF Statement Entries for Synchronized Files

Run Name (1-8)

Required entry

Enter the name specified in positions [1-8] of the RC statement.

Statement Type (9-10)

Required entry

Enter RF.

File Name (11-18)

Required entry

Enter the name of the file definition that specifies the fields and maps the data area for this file.

File Type (22-27)

Required entry

This entry determines the VISION:Transact qualifier to be associated with the file, and identifies the kind of data area being accessed. Enter FILE or FILEn (n=1-9) to associate the database or pseudo file with its qualifier. The qualifier for the database or pseudo file identified as FILE is blank or N, while the qualifier for databases identified as FILEn is the value of n (n=1-9). Enter WORKnn (nn=01-99) to define working storage (qualifier W). Enter PCBnn (nn=01-99) to associate the file with a PCB mask (qualifier V).

The PCB mask is used to map the PCBs. PCB01 addresses the first PCB of the program PSB (in IMS TM this is the I/O PCB); PCB02 addresses the second PCB, etc.

CICS: All working storage areas are placed in the TWA. Therefore, the total length of all work areas (work 01-99) must not exceed TWA size.

Synchronizing Field 1 (28-35)

This entry is required in an RF statement which specifies a database to be dynamically synchronized or to be both dynamically synchronized and screen synchronized. It specifies the name of the field containing the root key value for a direct read database, or the starting root key or generic value for a browse database. Leave this entry blank if the Screen Sync Type in position [53] is an M or U for a database that is not both screen synchronized and dynamically synchronized. This entry must be blank for RF statements specifying working storage or a PCB mask or pseudo files.

For a direct-read dynamically synchronized database, specify the name of the field containing the root key value of the desired database record. For a range browse database, specify the name of the field containing the starting root key value (lower limit) of the range; a different field name must be specified in Synchronizing Field 2. For a generic browse, specify the name of the field containing the generic key value (left-most portion of the key); this same field must be specified in Synchronizing Field 2.

The specified field may be in another database, in the PCB mask, in working storage, or it may be a screen field, a temporary field, or a flag field. Recursive synchronization is not allowed; e.g., if the synchronizing field for database 1 is in database 2, the synchronizing field for database 2 may not be in database 1. The source of the synchronizing field is specified in Sync Qualifier in position [52].

Synchronizing Field 2 (36-43)

This entry is valid only for range or generic browse synchronization. Leave the entry blank if Screen Synch Type M or U is specified and the SSDB is not to be browsed. Also this entry must be blank on an RF statement specifying working storage or a PCB mask or a pseudo file.

This entry specifies the name of the field containing the ending root key value (higher limit) of the range browse. For a range browse file, do not specify the same field that is specified for Synchronizing Field 1. For a generic browse, this field name must be the same name specified in Synchronizing Field 1.

The synchronizing field location is specified in the Sync Qualifier entry (position [52]). The specified field may be in another database, in the PCB mask, in working storage, or it may be a screen field, a temporary field, or a flag field. Recursive synchronization is not allowed; e.g., if the synchronizing field for database 1 is in database 2, the synchronizing field for database 2 may not be in database 1.

Sync Qualifier (52)

The presence of this entry identifies FILE or FILEn as a dynamically synchronized database. Note that dynamic synchronization only synchronizes the file on the primary root segment key. Thus, if multiple root segment keys are defined, only the primary key (key field 1) is used for direct or browse dynamic synchronization.

To specify the location of the synchronizing field, enter one of the following values:

Entry	Result
N	Field from the database or pseudo file whose File Type is FILE.
1-9	Field from the database or pseudo file whose File Type is FILE1-FILE9.
S	Field from the input screen.
T	Temporary field (default if an entry is made in [28-35]).
V	PCB mask.
W	Working storage.
F	Flag field.

When the synchronizing fields are in a screen or database segment, both Synchronizing Fields 1 and 2 must reside in the same segment. Synchronizing Field 2 must be located in the same area (PCB mask, working storage, etc.) as Synchronizing Field 1.

Screen Sync Type (53)

The presence of this entry identifies FILE or FILEn as a screen synchronized database. Leave this entry blank for a dynamically synchronized database (where a Sync Qualifier is specified) unless the database is both dynamically and screen synchronized.

Enter one of the following codes for a screen synchronized database:

Entry	Result
M	Database is screen synchronized and may not be updated.
U	Database is screen synchronized and may be updated.
P	Database is a pseudo file. A pseudo file may not be specified as dynamically synchronized.

DL/I Path Call Usage (54)

This entry controls the use of DL/I path calls by VISION:Transact for screen synchronized databases.

Enter one of the following codes:

Entry	Result
Blank	If APAR TR00115 was applied, the result is the same as for the I entry (below). If APAR TR00115 was not applied, the result is the same as for the A entry (below).
A	Use path calls whenever possible for this DL/I database, including insert calls.
I	Suppress all path insert calls for this DL/I data base. Path calls will only be used for segment retrieval and replacement. Path insert calls must be suppressed when one of the segments in a path is an LCHILD.
N	Do not ever use path calls for any DL/I database calls.

Record Length Field Name (55-62)

This entry is used to specify the name of the field to be used as the record length field for a VSAM variable length file. A record length field is required for a VSAM variable length file when the file does not contain variably occurring lower level segments and records are to be inserted or updated on the file or the length of a retrieved record needs to be examined.

The name specified must be unique within the file definition for this file. When referring to the record length field in the application, the file qualifier, as defined by the File Type entry on the RF statement, is used.

A record length field may only be specified for VSAM KSDS and ESDS file types (record format K or E). VSAM RRDS files (record format R) can only have fixed length records. A record length field must not be specified if variably occurring segments are defined in the file definition (i.e., segment count fields must not be present).

The record length field is a 4-byte fixed point binary field. After a read operation, the field contains the length of the record that was retrieved. Programmers must always supply a value for the record length field on insert operations. For update operations, the length need only be modified if the existing record length needs to be changed. Note that the length of records can only be modified in VSAM KSDS files. The length of records cannot be altered in VSAM ESDS files.

VISION:Transact verifies the contents of the record length field prior to an insert or update operation. If the value of the record length field is less than or equal to 0, greater than 32,767, or greater than the length of the record as defined in the file definition, then the input screen is redisplayed with message S04 indicating that a record length error has occurred.

Storage Type (64)

This entry applies only to working storage or pseudo file records. Leave this entry blank or enter R to indicate that this data area is to be re-initialized at the start of each application interaction (i.e., the start of each input screen processed; this does not include Recycle interactions). Enter S to specify that this work area or pseudo file is to be saved between interactions. Only fixed length pseudo files (FD record format F) can be saved between interactions.

When S is entered, more than one WORKnn or pseudo file may be saved in the Scratch Pad Area (SPA, under IMS TM) or on Temporary Storage (under CICS TS). Working storage areas are saved in ascending sequence (e.g., WORK01, WORK03, etc.) followed by any pseudo file records also saved in ascending sequence (e.g., FILE, FILE1, etc.).

IMS TM: Working storage and pseudo files reside in a separate GETMAIN area. When S is entered in Storage Type, the working storage areas and pseudo file records identified on RF statements with the S entry are saved in and retrieved from the SPA, thereby making the generated application conversational. The total length of all saved areas must not exceed 32,740 bytes.

CICS: When S is entered in Storage Type, the working storage areas and pseudo file records identified on RF statements with the S entry are saved in and retrieved from CICS Temporary Storage. The data is saved between interactions in Temporary Storage.

CICS: All working storage areas (regardless of their Storage Type entry) are placed in the TWA (pseudo files are not in the TWA). Therefore, the total length of all working storage areas (WORK01-WORK99) must not exceed the maximum TWA size.

Authorization ID (65-72)

For fully qualified table and view names, enter the authorization ID of the creator of the tables or views for any whose authorization id is not provided on the LS statement. If the authorization id has been provided on all LS statements within the definition, the value supplied here will not be used.

If an unqualified table or view name or synonym name is to be used for any of the segments within the file, then this entry must be left blank. In this case, the LS statement must provide the creator or authorization ID for all table and view names which are to be fully qualified.

RP Statement - Run Parameter

In an application generation run, definition run, or source statement retrieval run the RP statement supplies a parameter for the run. One parameter can be specified per RP statement.

Run Name (1-8)

Required entry

Enter the name specified in positions [1-8] of the RC statement.

Statement Type (9-10)

Required entry

Enter RP.

Parameter Name (11-18)

Required entry

Enter one of the following parameter names:

Entry	Result
FASTPATH	To generate an IMS TM application to process fast path databases. The IMS TM application must be non-conversational, each input screen must output only one screen to the I/O PCB or an alternate response PCB, and the RC statement must specify multiple messages. Enter no parameter value (APPGEN run only). This parameter is not allowed under CICS.
LKEDCALL	To name a program to be link edited with the application. The parameter value names the program (APPGEN run only).

Entry	Result
MANUAL	To print a Design Manual to summarize screen flow, data usage, and provide cross-references. Enter no parameter value (APPGEN run only).
MULTILIB	To override the primary library (default=M5LIB), to alter the search order of secondary libraries (default=M5LIB1-M5LIB9), or to disable alternate libraries. The first library named in a MULTILIB parameter overrides the primary library. Subsequent MULTILIB parameter values, in order, add a secondary library to the search sequence. The parameter value may be M5LIB or M5LIBn (n=1-9).

Parameter Value (19-71)

For a parameter entry of FASTPATH or MANUAL, leave this entry blank. For the parameter LKEDCALL, name the program to be link edited with the VISION:Transact application in positions [19-26]. For the parameter MULTILIB, enter M5LIB or M5LIBn (n=1-9) in positions [19-24].

SC Statement - Screen/COMMAREA Control

In an application generation run only, use a screen/COMMAREA control (SC) statement to describe the flow of control from one screen or COMMAREA section to another or from one application to another. It can also be used to include special procedural requests to assist in the processing of particular screens or COMMAREA sections.

OPER	LOGIC LEVEL	CONN	OPER A Q, FIELD	OPER B Q, FIELD	OUTPUT SCREEN	OPER C Q, FIELD	PART FIELD
bb	N	Y/R	N,N	N,N	R	Y	Y ³
Relational (EQ, NE, GT, LT, GE, LE)	Y	Y/R	R,R	R,R	Y	Y	Y
OC	N	N	N,N	N,R	N	N	N
OS	N	Y ¹	N,N	Y,Y ²	R	Y	Y ³
XR	N	N	N,N	R,R	N	N	N
SG	N	N	Y,R	N,R ⁴	N	N	N

Y = This entry may be specified.
R = This entry is required.
N = This entry is prohibited.
Y/R = This entry may be specified or required based on the value of the CONREQ parameter in M5PARM.

1. E is the only valid connector.
2. O is the only valid qualifier.
3. C is the only valid partial field operand.
4. SELECT ALL is the only valid entry for Operand B.

Figure 2-4 Summary of SC Statement Entries

Input Screen or COMMAREA Section Name (1-8)

Required entry

Enter the name of the input screen or COMMAREA section. All SC statements naming a given input screen or COMMAREA section in an application generation must be grouped together.

Statement Type (9-10)

Required entry

Enter SC.

Statement Sequence (11-13)

This entry is ignored. SC statements are always processed in input sequence.

Logic Level (14)

If this SC statement has an O (OR) or A (AND) logic connector in position [15], enter its logic level (1-9). Leave this entry blank and a logic level of 0 is assumed. If no logic connector is specified, implicitly or explicitly, leave this entry blank.

The logic level number (1-9) equals the number of pairs of parentheses needed to enforce the priority of the logic connector. To determine the logic level, follow these steps:

1. Write the conditions of the logical expression using ANDs, ORs, and parentheses, where appropriate. This expression may eventually be coded using multiple SC statements.
2. Count the pairs of parentheses surrounding the connector associated with this SC statement. Enter this count as the Logic Level [14] of this SC statement.

If logic levels of several related logic connectors are the same or are not specified, AND takes precedence over OR. For example, A OR B AND C is interpreted as A OR (B AND C).

Connector (15)

Enter A for AND, O (alpha O) for OR to join the relational operation (EQ, NE, GT, LT, LE, or GE) on this SC statement with the one on the previous SC statement. If the CONNECTOR entry is blank and CONREQ=N is specified in M5PARM, the default OR connector is assumed. The CONNECTOR entry cannot be blank when joining two or more relational statements if CONREQ=Y is specified in M5PARM. See the *Advantage VISION:Transact for CICS Installation Guide* for more information on M5PARM.

The AND logical connector has precedence over the OR connector. Use the AND connector to join coincident logical expressions, and use the OR connector to join alternative logical expressions.

Enter E for ELSE to indicate that if this SC statement is processed, it serves as a logical default to previous SC statements for the same input screen or COMMAREA section. In other words, if the preceding relational expression(s) evaluate as false for the same input screen or COMMAREA section, then the ELSE condition is evaluated.

This entry must be blank when this SC statement or the previous one has no relational operation. The exception is the E connector which may be specified with an OS operator or when the operator is blank.

Operand A (16-24)

Enter the qualifier and the name of a field used for comparison (EQ, NE, LT, GT, GE, or LE operators) or the segment to which the SG operator applies. Leave this entry blank if the Operation in positions [25-26] is blank or if the operation is OC, OS, or XR.

Qualifier A (16)

If the operator is relational, enter one of the following qualifiers:

Qualifier	Source of Field
T	Temporary field.
F	Flag field (VISION:Transact system field).
W	Working storage area.
V	PCB mask field.

If the operator is the segment operator (SG), enter one of the following qualifiers:

Qualifier	Source of Field
Blank/N	Database or pseudo file whose File Type is FILE.
1-9	Database or pseudo file whose File Type is FILE1 to FILE9.

Field Name A (17-24)

If the operator is relational, enter the name of the field as defined in a file definition, temporary field definition, or as a system flag to be used for comparison. If the operator is a segment operator (SG), enter the name of the segment as it appears on the LS statement in the file definition.

Operation (25-26)

Enter an input control operator, a relational operator, an output control operator, or a segment operator. If you leave this entry blank it is similar to using the OS operator.

An input control operator (XR) must precede all other SC statements for the input screen or COMMAREA section and is used to specify the input requests or procedures (type 1, 2, or M) to be executed for the input screen and their order of execution.

A relational operator is used to compare Operand A and Operand B. The comparison returns a TRUE or FALSE condition. If the condition is TRUE, this SC statement and the following ones prior to a subsequent ELSE connector determine the output screens or COMMAREA sections and special output processing to be performed. If the condition is FALSE, the conditions specified on the subsequent ELSE connector is evaluated. You can combine relational operations of contiguous SC statements using AND and OR connectors.

The segment operator SG is used to apply the segment function indicated in Operand B to the screen synchronized or pseudo file segment named by Operand A. For instance, the SG operator with the segment function SELECT ALL releases the segment named in Operand A and all its children from the intermediate structure for output processing. The SG operator must precede all output screens or COMMAREA sections to which it applies. An SG operator that appears before all output selection statements for an input screen or COMMAREA section applies to all output screens that are generated as a result of the input screen or COMMAREA section. If the SG operator appears after output selection statements, then it applies only to the selected output screen or COMMAREA section.

An output control operator is used to specify a type O request or procedure, the output screen or COMMAREA section, and its destination, and indicate any special output characteristics for this interaction. The output characteristics include: VISION:Transact indicator suppression, screen recycling, conversational processing, transaction initiation, and output page limitations.

Enter one of the following operators:

Operator	Description
Input Control Operator	
XR	Used to specify the names of type 1, 2, or M requests or procedures that are to be executed for the input screen or COMMAREA section and to specify the order in which they are to be executed. The Operand B qualifier contains the type (1, 2, or M) and Operand B names the request or procedure.
Relational Operators	
EQ	Operand A equals Operand B or matches the pattern in Operand B.
NE	Operand A does not equal Operand B or does not match the pattern in Operand B.
LT	Operand A is less than Operand B.
GT	Operand A is greater than Operand B.
LE	Operand A is less than or equal to Operand B.
GE	Operand A is greater than or equal to Operand B.

Operator	Description
Segment Operator	
SG	Use with an Operand B of SELECT ALL to release the segment named in Operand A and its children from the intermediate structure for output processing.
Output Control Operators	
OS	Used to specify the type O request or procedure (if any), the output screen or COMMAREA section, and its destination. Operand B specifies a request or procedure to be executed on output (type O). Operand C can be used to supply an output destination for the screen. See Operands B and C.
OC	Use to control output interaction functions. Operand B specifies a function to be used to suppress VISION:Transact indicators, specify conversational transactions, start a CICS transaction, recycle the output screen, or limit the number of output pages. See Operand B.

If a request or procedure is referenced by an XR or OS operator, its ER statement must have blanks in Screen Name. If the named item is a request and it is not found in the APPGEN input, it will be retrieved from the library.

Operand B (27-49)

Operand B identifies the field, constant, or named pattern to compare with Operand A, or it names the request/procedure or parameter for special segment or control operators. Leave Operand B blank if the Operator entry [25-26] is blank.

Type of Operator	Role of Operand B
Input control	Specifies the input request/ procedure type and name.
Relational	Logically compared with Operand A.
Output Control	Specifies output control function.
Segment Operator	Names the SELECT ALL function.

Operand B Qualifier (27)

Enter one of the following codes:

Entry	Result
Blank	With SG or OC operator, or to execute no type O request or procedure for the screen or COMMAREA section named by an OS operator.
C	Constant character string field (relational operators only).
D	Decimal constant field (relational operators only).
F	Flag field (relational operators only).
P	Pattern (EQ or NE operators only).
T	Temporary field (relational operators only).
V	PCB mask field (relational operators only).
W	Working storage area field (relational operators only).
1, 2, M	Used with XR operator to name the input request or procedure type 1, 2, M.
O	Used with OS operator to name a type O request or procedure.

Operand B (28-49)

If the operator is:

EQ,NE and Operand A is to be checked for a valid date, enter one of these date patterns (ensure a period is in position [28] and a P in position [27]):
.MMDDYY .MMYYDD .DDMMYY .DDYYMM .YYMMDD
.YYDDMM .TODAY .JULIAN

Refer to [Chapter 3, “Advantage VISION:Transact Flags and Patterns”](#) for more on date patterns.

and Operand A is to be checked for a numeric pattern, enter one of these numeric patterns (ensure a period is in position [28] and a P in position [27]):

.NUMERIC .INTEGER .FLOAT

Refer to [Chapter 3, “Advantage VISION:Transact Flags and Patterns”](#) for more on numeric patterns.

otherwise, enter a constant or the name of a field to compare with Operand A.

GT,GE, Enter a constant or the name of a field to be compared with Operand A.
LT,LE

SG Enter SELECT ALL to release the segment named by Operand A and its children from the intermediate structure for output processing.

XR Enter a type 1, 2, or M request or procedure name to be executed for the input screen or COMMAREA section. The ER statement must not name a screen or COMMAREA section. Within each request/procedure type, the requests/procedures are executed in the order they are named by XR operators on SC statements. If the named item is a request which is not instream, VISION:Transact searches the libraries.

OS Enter a type O request or procedure name to help build the output screen or COMMAREA section named on this SC statement. The ER statement must not name a screen or COMMAREA section. If Operand B is blank, no output request or procedure is executed for the named output screen or COMMAREA section even though one may exist in the input stream for the named screen or COMMAREA section. If the named item is a request which is not instream, VISION:Transact searches the libraries.

OC Enter one of the following values to control output processing:

XIND	Suppresses the VISION:Transact indicators used for missing, invalid, and numeric overflow fields. This is useful when the screen is output to a program rather than a terminal.
CONV (IMS TM)	Inserts the SPA in the message queue before the first output message allowing the conversation to be continued by the next transaction (the transaction named in Operand C of the preceding SC statement).
CONV (CICS)	Enables the application to converse with the primary terminal. The application sends the output screen to the terminal and waits for a response. Such conversations should be used with caution, since they can adversely affect overall response and resources.
START (CICS)	Initiates a transaction for the transaction contained in the TRANCODE flag to run asynchronously, and to pass one page of output to it. The started transaction is associated with the originating terminal or the terminal named in Operand C of the preceding SC statement. The output screen and all saved working storage and pseudo files (Storage type S) are passed to the started transaction.
RECYCLE	Outputs and re-inputs one page of the output screen named on the preceding SC statement, bypassing any destination. This must be an input screen of the application and no other output screens can be processed with it. Ensure there is an exit from this recycle loop. VISION:Transact terminates after 10 consecutive RECYCLES. Temporary fields, working storage, and pseudo files are not re-initialized and retain their values during recycle processing.
OUTLIM nnn	...Limits or specifies the number of pages produced by an output screen. Right-justify the value of nnn which can range from 001 to 999. This entry overrides the Output Control entry in the SD statement. An OUTLIM value greater than 1 causes the application to build and temporarily store a set of output screens. You can "page" through this set of screens using MFS or BMS paging. See Chapter 7 of the <i>Advantage VISION:Transact for CICS Environment Guide</i> for more information on MFS and BMS features supported by VISION:Transact generated applications.

Output Screen or COMMAREA Section Name (50-57)

Enter the name of the screen or COMMAREA section, as specified on the SD statement, to be displayed (output) if the logical expression is true. (Refer to the note under the connector entry.) Any subsequent logical expression for the same input screen or COMMAREA section must begin with the ELSE connector.

The output screen or COMMAREA section may be subsequently input to this or another application. The destination application need not have been implemented with VISION:Transact.

Additional output screens or COMMAREA sections selected by the same logical conditions are specified on subsequent SC statements containing OS operators or only an input screen name or COMMAREA section, output screen name or COMMAREA section, and destination. Only one output screen or COMMAREA section from one interaction may be directed to the originating or primary terminal.

Leave this entry blank if the Operator is XR, SG, or OC.

Operand C (59-67)

This entry specifies a field or constant whose value is the name of the destination to which the output screen is to be sent. Enter either a field name or a character constant, or leave this entry blank if the screen is to be output to the same terminal from which the interaction originated. Leave this entry blank if the operator is XR, OC, or SG, or if the operator is OS and the next SC statement has an OC operator with a RECYCLE function.

IMS: Operand C may specify an LTERM or transaction as the destination for the output screen.

CICS: Operand C may only specify a terminal ID as the destination of an output screen. If an OC operator with the START parameter follows, the transaction to be started is named in the TRANCODE flag. The transaction is started for the originating terminal (if Operand C is blank) or the terminal named in Operand C.

Qualifier C (59)

This entry indicates the source of the field or constant used to designate an alternate destination. Enter one of the following codes:

Entry	Result
C	Character string constant field.
F	Flag field.

Entry	Result
T	Temporary field.
V	PCB field.
W	Working storage field.

Field Name/Constant (60-67)

This entry specifies the field name or character constant whose value is the name (LTERM or TRAN code) of the destination.

IMS: The destination may be a terminal or the transaction code of an application.

CICS: The destination may only be a terminal.

Partial Field (68-72)

These entries allow the processing of part of a character string field. The partial field specifications may be used with fields defined with field type C. Leave this entry blank if the operator is SG, XR, or OC.

Starting Position (68-69)

Enter the location of the 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 LSTART flag.
LN	Current value of LNUMBER flag.
MS	Current value of MSTART flag.
MN	Current value of MNUMBER flag.
PS	Current value of PSTART flag.
PN	Current value of PNUMBER flag.
RS	Current value of RSTART flag.
RN	Current value of RNUMBER flag.

Number of Characters (70-71)

Enter the number of characters forming the partial field.

Entry	Result
Blank	No partial fielding.
1-99	Number of characters forming the partial field (not exceeding the length of the field).
LS	Current value of LSTART flag.
LN	Current value of LNUMBER flag.
MS	Current value of MSTART flag.
MN	Current value of MNUMBER flag.
PS	Current value of PSTART flag.
PN	Current value of PNUMBER flag.
RS	Current value of RSTART flag.
RN	Current value of RNUMBER flag.

Operand A, B, or C (72)

Identify the Operand to which the partial field specifications apply.

Entry	Result
A/Blank	Operand A.
B	Operand B.
C	Operand C.

OS operators may only specify Operand C.

SD Statement - Screen/COMMAREA Definition

The SD statement is used in a definition run or an application generation run to begin a screen definition. This statement provides specifications that are pertinent to the entire screen processing definition.

Screen or COMMAREA Section Name (1-8)

Required Entry

This entry names the screen or COMMAREA section as it is defined to VISION:Transact. Enter a unique screen or COMMAREA section definition name.

IMS TM: If the Screen Identifier [25-31] is blank, positions [1-5] of this entry must provide a unique name to MFS. This entry is used to name a MID, MOD, and DIF/DOF. VISION:Transact takes the MOD name from positions [1-5], appends an I to the MOD name for the MID name, and appends an F to the MOD name for the DIF/DOF name.

CICS: VISION:Transact takes the MAP name from positions [1-7] of this entry. Also, if the Screen Identifier [25-31] is blank, positions [1-7] of this entry must provide a unique name to BMS, since this entry is used to name the MAPSET. If necessary, VISION:Transact modifies the MAP name internally using positions [1-6] so its not the same as the MAPSET name. The MAP name must be unique within the MAPSET.

Statement Type (9-10)

Required entry

Enter SD.

Catalog Action (11)

To catalog, replace, or delete a screen definition from the library, enter one of the following codes:

Entry	Result
Blank, A, or N	Add the screen definition to the library if it is not yet cataloged.
D or Y	Delete the screen definition from the library.
R	Replace the screen definition if it is already cataloged; otherwise add it.

Note: The library must be opened for update during an APPGEN run in order to catalog, replace, or delete a screen from the library (see the Library Open specification in M5PARM and on the RC statement).

Glossary (12)

This entry specifies the type of glossary listing. Enter one of the following codes:

Entry	Result
Blank/A	Output an image of the screen followed by a listing of all fields on the screen in order of screen location.
L	Output an image of the screen only. No fields are listed.
N	No glossary.

Field Designator (13)

This entry specifies the character used on the Screen Layout (SL) statements to designate the location of the variable data fields which are entered as input or displayed as output by the application. Enter any non-blank character, or leave blank to use the default field designator in M5PARM. (See [SL Statement - Screen/COMMAREA Layout on page 2-78](#).) VISION:Transact is delivered with the underscore (_) character as the default field designator in M5PARM.

Highlight Designator (14)

This entry specifies the character used to designate highlighting of literal text on the Screen Layout (SL) statements. Leave this entry blank if the default highlight designator is to be used. Enter any non-blank character, different than the field designator, to serve as the highlight designator. The highlight designator must enclose any literal text to be highlighted. VISION:Transact is delivered with the "at" sign (@) character specified as the default highlight designator in M5PARM.

Automatic Format (15)

This entry specifies whether VISION:Transact is to provide an automatic screen layout. Enter one of the following codes:

Entry	Result
Blank/N	Position of all fields in the screen definition is specified via SL statements.
C	VISION:Transact automatically provides a columnar format for all screen fields that are not explicitly placed on the screen layout by an SL statement.

Upon request, the system automatically provides a display format for the entire screen. You can also provide a layout for part of the screen; a format for the rest of the screen can be provided by VISION:Transact. In either case, enter the S0-Sn statements for every field which is to appear on the screen.

If SL statements are also present, they specify the first portion of the screen layout. Any screen field defined by an Sn statement without a corresponding SL statement is automatically formatted on the screen in subsequent display lines.

The automatic format is columnar and each field on the screen is preceded by a literal field description. The field description is derived from Operand A unless you use a DS operator on the Sn statement. All field descriptions are padded to the right with periods until they equal the length of the longest description.

Normally, only one column is provided on the screen. If the number of fields to be displayed exceeds the lines on the screen, VISION:Transact attempts to format two fields on each display line. The amount of space reserved for a field and its descriptor is the column width, which either equals the whole screen width or half of it minus the length of the longest descriptor minus 2.

If the screen cannot accommodate all the specified fields in two columns, the screen definition is invalid. You must correct this by reducing the number of fields on the screen or provide a full screen layout.

Output Control (16)

This entry controls the building of multiple output pages for a screen on which to display all occurrences of selected data. Enter one of the following codes:

Entry	Result
Blank/S	Suppresses displaying data for a given screen segment within its current parent segment once the number of occurrences specified on the SS statement are reached. Only one screen page is output.
A	Displays all occurrences of the selected data on one or more screen pages.

Note: An entry of A causes the application to build and temporarily store a set of output screens. You can “page” through this set of screens using MFS or BMS paging. See Chapter 7 of the *Advantage VISION:Transact for CICS Environment Guide* for more information on MFS and BMS features supported by VISION:Transact generated applications.

You can control the number of pages generated through the use of the OUTLIM parameter of the OC operator in screen control. This parameter can also be used to override the Output Control entry of blank or S.

Input Transaction Code (17-24)

This entry names the transaction to process the screen on input. This entry is useful when the screen is always input to a particular transaction.

If the screen is not input to any transaction (e.g., an output only screen), or if the TRANCODE flag is used to determine the inputting transaction, leave this entry blank (see Chapter 3).

IMS TM: Enter the name of the transaction defined in the IMS TM TRANSACT macro that inputs this screen. For VISION:Transact applications the transaction name is the Run Name/Transaction Code entry [1-8] on the RC statement of the application generation run.

IMS TM: If a TRANCODE flag is to name the input transaction, it must be defined as a source and destination field on an S0 statement in this screen definition. Note if this entry is left blank, the TRANCODE flag must be defined on an S0 statement. If this entry is specified *and* flag TRANCODE is defined on an S0 statement in the screen, then this entry provides the default input transaction code when the TRANCODE screen field is left blank.

CICS: Enter the name of the transaction identifier defined in the CICS Program Control Table (PCT) that inputs this screen. VISION:Transact takes the transaction identifier from positions [1-4] of the Run Name/Transaction Code entry on the RC statement of the application generation run.

Screen Identifier (25-31)

IMS TM: In the MFS generated source statements, this entry is used by VISION:Transact to name the MID, MOD, and DIF/DOF for this screen. If this entry is seven characters, VISION:Transact names the MOD with it, appends an I to it and names the MID, and uses the first six characters to name the DIF/DOF. If this entry is six characters or less, VISION:Transact uses the first five characters to name the MOD, appends an I to it to name the MID, and appends an F to it to name the DIF/DOF.

IMS TM: If this entry is left blank, the first five characters of the Screen Name are used. This name must be unique to MFS.

CICS: In the generated BMS source statements, this entry is used by VISION:Transact to name the MAPSET for the screen. If multiple screens are defined in a run, the first screen's identifier is used to name the MAPSET which will contain one map for each screen defined in the run.

CICS: If this entry is left blank, the first seven characters of the Screen Name (positions [1-7]) name the MAPSET.

Null Segment Processing (35)

To control how the application processes null screen segments, enter one of the following codes:

Entry	Result
Blank/N	Ignore all null screen segment occurrences.
A	Process all null screen segment occurrences.
B	Process all null screen segment occurrences in type 1 requests or procedures. Ignore segment occurrences still null after type 1 requests or procedures are processed.

Device ID (36-43)

This entry indicates the type of terminal for which the screen definition is being generated. The device name must correspond to one of the devices or device groups in the M5PARM Device Table. These are listed on the VISION:Transact signon page. If this entry is left blank, the M5PARM default device is assumed.

You can specify the term COMMAREA for this entry to indicate that the screen being defined is intended for a COMMAREA program only, and any constraints unique to 3270 Terminal programs do not apply. No BMS mapping statement will be generated when the Device ID entry contains COMMAREA.

SL Statement - Screen/COMMAREA Layout

The SL statements are used in a definition run or application generation run to define the layout of a screen display.

An SL statement specifies the literal text, the position and length of data fields, and optionally, the highlighting, color, and special characters composing the literal text.

Each field specified in an SL statement must correspond to at least one S0 statement that provides its characteristics.

One line of the screen must be reserved for error messages to the terminal user. By default, this is the last line of the screen. However, you may place this line anywhere on the screen by using the MSGLINE flag field. For details on how to do this, see Chapter 5 of the *Advantage VISION:Transact Reference Guide*.

If SL statements are present when the SD statement Device ID entry specifies COMMAREA, the only function that the SL statements serve is to specify the length of a field in the COMMAREA through the presence of a field designator sequence for a field.

Screen Name (1-8)

Required entry

Enter the same screen name as specified on the SD statement.

Statement Type (9-10)

Enter SL.

Line Number (11-13)

Enter a number to relate this SL statement to a particular line on the screen. If this entry is blank, VISION:Transact assumes a line number that is 1 greater than the previous SL statement, and if this is the first SL statement, a 1 is assumed. Screen lines with no corresponding SL statements are left blank, except when Automatic Formatting is specified on the SD statement.

Continuation (14)

Enter one of the following codes to indicate which portion of the screen line that the TEXT on this SL statement specifies:

Entry	Result
Blank/0	Text defines columns 1-57 of this screen line.
1	Text defines columns 58-114 of this screen line.
2	Text defines columns 115-171 of this screen line.

If, on a particular screen line, one group of columns (1-57, 58-114, or 115-171) are not specified by an SL statement, that group is assumed to be blank.

Note: The number of valid continuations is determined by the screen width specified for the particular device ID in the terminal table in M5PARM; this information is shown on the signon page. If the screen is narrower than the SL statement Text, continuations are invalid. If the screen is wider than the SL statement Text, the maximum number of continuations is calculated. It is possible for an SL statement to be truncated. For instance, a screen that is 132 characters wide allows a maximum of two continuation statements. If the second continuation is used, only its first 18 positions are valid.

XATTR (15)

This entry works with Text entries to set the extended attributes available on some 3270 terminals: color, extended highlighting (reverse video, blinking, or underlining) and programmed symbols. An XATTR entry specifies that some data field or literal on this screen line has an extended attribute set.

Enter one of the following codes:

Entry	Result
Blank	This SL statement describes the screen layout or format for this screen line (literals and data fields).
C	Color attributes are specified for some data fields or literals on this screen line.
H	Reverse video, blinking, or underlining is specified for some data fields or literals on this screen line.
P	Programmed symbols are specified for some data fields or literals on this screen line.

See the Text specification (below) for the attribute codes.

Text (16-72)

This entry describes the format of a screen line or the extended attributes applied to data fields or literals on this line. Enter the literal text and fields as they are to appear on the screen according to the following conventions (XATTR entry must be blank):

1. Line 1, position 1 can only contain a blank or the highlight designator.
2. Enter literal text as it is to appear on the screen. A literal string which is a series of contiguous non-blank characters cannot be more than 248 bytes.
3. To highlight any literal text, enter one highlight designator before it and one after it.
4. Indicate data fields with a series of field designators to specify the position and length of the field. Each data field is limited to 255 field designators.
5. Data fields may be continued on successive lines; in this case, field designators occupy the end of the preceding line and the beginning of the succeeding line.
6. Each series of field designators (one data field) must be preceded and followed by a blank or a highlight designator.
7. To move the error message from the last line of the screen, use the MSGLINE flag field. On the chosen line, enter enough field designators to display the message. To avoid truncating any messages, use at least 79 field designators. The corresponding S0 statement must be in the root of the screen and it must name the flag field MSGLINE and must be a non-virtual field. If the flag field MSGLINE is not referenced in an S0 statement, the message line defaults to the last line of the screen. See Chapter 5 of the *Advantage VISION:Transact Reference Guide*.

To describe the extended attributes on an SL statement, write a string of codes for the desired attribute in the same positions of the XATTR SL statement as the literal or data field occupies on its SL statement. For example, if the literal ENTER is coded in positions 30-34 of the SL statement in line 2, and you want it to display in red, code another SL statement for screen line 2 with a C in XATTR and a string of Rs in positions 30-34. There are different entries in Text depending on the entry you made in XATTR.

XATTR Entry	Attribute Code	Result
C (color)	N	Neutral.
	D	Device default.
	B	Blue.
	R	Red.
	P	Pink.
	G	Green.
	T	Turquoise.
H (extended highlight)	Y	Yellow.
	D	Device default.
	B	Blinking.
	R	Reverse Video.
P (programmed symbols)	U	Underlined.
	E	Extended Graphics (EGCS). Fields must start in an oddnumbered column and must be an evennumber of characters In length.
	Any other character	Installation dependent.

Attribute codes for a literal or data field on one XATTR line must not be mixed. For example, a color attribute of BBBYBB for one data field is illegal.

For more examples of coding extended attributes, see Chapter 5 of the *Advantage VISION:Transact Reference Guide*.

Sn Statement - Screen/COMMAREA Content

In an application generation run or a definition run, use Sn statements (n=1 to 9) to code multiple lines of specifications for an input screen field. These specifications can consist of the following:

- Edit/Validation.
- Application-specific messages (to the terminal user) if the input field value does not satisfy validation criteria in the preceding S0 and Sn statements.
- Application-specific messages that override VISION:Transact messages when specific error conditions apply (for example, input required, or segment not found).
- Dynamic selection of segment and field actions.
- Assignment of names to screen fields.
- Assignment of lengths to virtual screen fields.
- Default data values.
- Descriptive text for fields when automatic formatting is used.

A set of Sn statements for a screen field must accompany an S0 statement. Because screen definition statements are processed in input sequence, Sn statements must immediately follow the S0 statement for the screen field to which they apply. Also, since individual statements in a stored screen definition cannot be updated, you need not uniquely identify each Sn statement. Therefore there is no distinction among the different values of n (1-9). For example, two additional lines of processing may be identified by one S1 and one S2 statement, or by two S2 statements. You can use as many Sn statements as you need.

Sn statements are not allowed for IMS TM passwords.

Screen Name (1-8)

Required entry

Enter the same screen name as specified on the SD statement.

Statement Type (9-10)

Required entry

Enter Sn (where n=1 to 9). Within each screen field you may repeat each value of n as many times as necessary.

Screen Segment Number (11-12)

Leave this entry blank.

Virtual/Same/After (15)

Leave this entry blank.

Operand A (16-24)

Leave this entry blank.

Action (38)

Use this entry only when the Edit/Validation operator is IF. It must specify the action that is to apply for the input field when the relation specified by Operand B is true.

When dynamic actions (IF operators) are specified for an S0-Sn set, specify the most likely logical condition/action on the first Sn statement since it is evaluated first. The action specified for the S0 statement (which is input first) should be the least likely (though it is the default) since all IF conditions must be evaluated as false before the default action is used. If the S0 statement specifies a segment action, so must all its associated Sn statements. If the S0 statement specifies a field action so must all of its associated Sn statements. In other words, segment and field actions must not be mixed within an S0-Sn set.

Enter one of the following codes as actions to be performed when the relational test specified by Operand B is true.

Entry	Segment Action
D	Delete the existing segment. (Do not use this action on the root segment of an ISAM or VSAM-ESDS file).
I	Insert a segment.
M	Retrieve (Match) the segment but do not update it.
Q	Repeat the action used on the parent of this segment.
U	Retrieve and update an existing segment.
V	If the segment exists, retrieve and update it; if it does not exist, insert it.
X	Perform no segment action (ignore it).

Entry	Field Action
A	Add the value of the screen field to the value of the destination field.
B	Blank or zero the destination field.
N	Perform no action (null).
P	Unconditionally replace the value of the destination field with the value of the screen field.
R	If the screen field is not blank, replace the value of the destination field with the value of the screen field.
S	Subtract the value of the screen field from the value of the destination field and put the difference in the destination field.

For a more complete description of these actions, see [S0 Statement - Screen/COMMAREA Field on page 2-97](#).

Logic Level (39)

Use this entry only when the Edit/Validation operator is EQ, NE, LT, GT, LE, or GE and a logic connector is entered on this Sn statement.

Specify the priority of the logic connector on this Sn statement among a series of multiple AND or OR conditions. Enter a number from 0 (or blank) at the lowest priority through 9 at the highest priority to ensure that the logic connector on this Sn statement is evaluated at the proper time. This number serves the same purpose as parentheses in a logical expression.

To enter the correct logic level number, follow these steps:

1. Using ANDs, ORs, and parentheses, write the complete logical expression that will eventually span several S0-Sn statements.
2. In this expression, locate the logical connector (AND or OR) that corresponds to the logical connector on this Sn statement. Count the pairs of parentheses surrounding this connector.
3. Enter this count as a logic level on this Sn statement. If no parentheses surround this connector in the expression you may enter a 0 or leave position [39] blank.

Note: Remember when no logic level is entered and one is implied, VISION:Transact gives AND priority over OR.

Connector (40)

This entry indicates an AND or OR logical relationship between this Sn statement and the previous S0 or Sn statement.

If the Operator entries on this Sn statement and the previous S0/Sn statement are not EQ, NE, LT, GT, LE, or GE, leave this entry blank.

Enter O (the letter) to indicate that this validation operation is connected to the previous one by an OR logical operator. An OR implies that the logical condition on this Sn statement may be TRUE instead (an alternative) of the one on the previous S0/Sn statement.

Enter A to indicate that this validation operation is connected to the previous one by an AND logical operator. An AND implies that the logical condition on this Sn statement must be TRUE at the same time as the one on the previous S0/Sn statement.

VISION:Transact assumes that AND has priority over OR, unless a logic level indicates otherwise.

Edit/Validation Operator (41-42)

This entry specifies operations to be performed on the data in the screen field. Except for the field descriptor and edit specifications, it is only valid with input screen fields.

Enter one of the following codes:

Entry	Result
Logical Operators	
EQ	Input screen field is equal to Operand B or matches the pattern in Operand B.
NE	Input screen field is not equal to Operand B or does not match the pattern in Operand B.
LT	Input screen field is less than Operand B.
GT	Input screen field is greater than Operand B.
LE	Input screen field is less than or equal to Operand B.
GE	Input screen field is greater than or equal to Operand B.

Entry	Result
Other Operators	
DF	If the screen field is blank, operand B provides a default value.
DS	Operand B provides a character constant to describe this field when VISION:Transact provides an automatic format for the screen. See SD Statement - Screen/COMMAREA Definition on page 2-73 for details on automatic formatting.
EC	This operator works in conjunction with the ET operator of a contiguous Sn statement. Operand B of this Sn statement specifies the error condition whose VISION:Transact message is to be overridden by the B operand associated with the ET operator.
ED	Operand B specifies an edit mask that inserts characters in the screen field during output processing. If the screen field is also input, the edit mask defines the de-editing done on input before type 1 requests or procedures are processed.
EN	If the source field is numeric, EN inserts grouping characters among the numbers prior to output. The EN operator is invalid with a floating point source field. If the source field is character, no editing is done prior to output. However, when a non-blank screen field is input, EN right-justifies it and fills it with leading zeroes. If the screen field is blank, no editing is done. Operand B must be blank for the EN operator. The EN operator is ignored if the ED operator is present.
ET	This operator works in conjunction with the EC operator of a contiguous Sn statement. Operand B of this Sn statement specifies the text to override the VISION:Transact error message associated with the error condition indicated by operand B of the EC operator. Multiple, consecutive ET operators can be used to provide a longer message. Up to 79 characters are valid.

Entry	Result
IF	The control field specified in the Operand B positions [43-51] is logically compared, using an EQ or NE in positions [52-53], to the control field value specified in positions [54-67]. If the comparison is TRUE, the Action on this Sn statement is performed. If it is FALSE, the next IF operator is processed. If all IF comparisons fail, the Action specified on the S0 statement is performed. The control field must be a screen field or a flag field. If it is a screen field, it must be defined within the same screen segment as the IF operator, or in a parent screen segment. The same control field name should be used for each screen field to avoid the possibility of erroneous database updates, when multiple IF conditions are TRUE. The first IF condition that is TRUE determines the action to be performed; subsequent IF statements are not processed. Tests for alternate control field values must be specified by grouping IF operators. IF operators in an Sn set must not be interspersed with other operators. The connector specification cannot be used in an Sn statement that contains the IF operator.
LN	Operand B specifies the length of a virtual field, screen field, or COMMAREA field. If a destination is not specified for an input virtual field, LN is a required entry. If a destination is specified with no LN operation, the length of the virtual field is the same as the output edit length of the destination field. A virtual field is a character string field with a maximum length of 255 bytes. If an LN operation is specified for a non-virtual field and a corresponding field designator sequence is present on an SL statement, the LN operand value must match the length of the corresponding field designator sequence. If no LN operation is specified for a COMMAREA screen and there is no corresponding field designator sequence present on an SL statement, the length of the COMMAREA field is the same as the output edit length of the Source/Destination Field.
MS	Provides a message to the terminal user when the previous validation fails. Multiple consecutive MS operators may be used to provide a longer message. Up to 79 characters are allowed. However, the message is truncated to screen width minus one when the screen is less than 80 characters long. If the input validation is specified but no MS operators are present, a system supplied message is output if the validation fails. The first MS operator must immediately follow the last input validation operator.

Entry	Result
NC	Operand B specifies an alternate name for this input screen field that can be up to 24 characters long. When specified, this name will be used in the COBOL statements that define the COMMAREA. If no NC operator is provided for a field, the NM operand is used. If neither an NC or an NM operator is provided, the Source/Destination Field name for the screen field is used as the COBOL element name. The NC operator can be specified even when the NM operator is also specified.
NM	Operand B specifies a name that can be up to eight characters for this input screen field (uniqueness is not verified). If an NM operator is not used for a screen field, it takes its name from the destination field. If duplicate names are assigned by NM operators for a screen input field, an error is issued.
SE	Use only with an input virtual screen field. This operator associates it with a non-virtual screen field named by Operand B (Qualifier B must be an S). If the virtual screen field is in error, the associated field's error handling is used.

Operand B (43-67)

Operand B serves many purposes. For Edit/Validation operators other than IF, it specifies the field name or constant for a relational test; or it specifies screen field names, message codes, edit masks, field lengths, default values, or descriptions for automatically formatted fields. With the IF operator, Operand B specifies a relational test with each alternative Action. Here Operand B names a control field in positions [43-51], specifies a relational operator in positions [52-53], and provides a control field value in positions [54-67].

Qualifier B (43)

This entry identifies the operand's qualifier (field, constant, or pattern) for Edit/Validation operators other than IF. For the IF operator, refer to the Control Field specifications, beginning with [Control Field Qualifier \[43\] on page 2-93](#).

Leave this entry blank when the Edit/Validation operator is DS, EC, ED, EN, ET, LN, or NM. Code an "S" entry for the SE operator.

Identify the source of the field used for Edit/Validation by entering one of the following codes:

Entry	Result
Blank/N	Field from file whose type is FILE.
1-9	Field from file whose type is FILE1-FILE9.
S	Another field in the same screen segment or in a parent screen segment.
T	Temporary field.
F	Flag field.
W	Working storage field.
C	Character constant.
D	Numeric constant.
P	Pattern for validation (EQ or NE operators only).

Edit/Validation Field (44-67)

This entry identifies the operand field name, constant, pattern, or edit mask used for Edit/Validation operators other than IF. For IF operators, refer to the Control Field specifications, beginning with [Control Field Qualifier \[43\] on page 2-93](#). If the operator is DS, Operand B must be a character constant. If the operator is MS, Operand B can only be a character type field; qualifier D or P is not valid.

If the Edit/Validation operand for this statement is:

Operator	Operand B
EQ,NE	Enter a constant, field name, or a validation pattern to validate the input screen field. See EQ/NE - Pattern Validation on page 2-90 for details on pattern validation.
LT,GT, LE,GE	Enter a constant or field name to validate the input screen field.
DF	Enter the default value to be assigned to this screen field.
DS	Enter the descriptive text to be displayed with the contents of this screen field.
EC	Enter an error message code to be translated by the ET operator. See EC/ET Operators - Error Condition Codes on page 2-92 for more details.

Operator	Operand B
ED	Enter an edit mask consisting of edit insertion characters and system delimiters. Editing characters are inserted in an output field and removed from in input field. See ED Operator - Edit Masks on page 2-91 for more details.
EN	Leave this entry blank.
ET	Enter the text for an error message (1-79 characters) that can be continued on another Sn statement. See EC/ET Operators - Error Condition Codes on page 2-92 for more details.
LN	Enter the length of the screen field, specified by this statement. Maximum length is 255 characters.
MS	Enter the text to compose the message displayed when validation fails.
NC	Enter the unique name (1-24 characters) to be assigned to this field in the COBOL definition of the COMMAREA.
NM	Enter the unique name (1-8 characters) to be assigned to this screen field.
SE	Enter the name of the non-virtual screen field. This field must be in the same screen segment as the virtual screen field, or in a parent of that segment (Operand B qualifier must be S).

EQ/NE - Pattern Validation

For pattern validation, the Edit/Validation operator must be EQ or NE. Enter a P in Operand B qualifier [43], a period (.) in position [44], followed by the pattern name.

To validate a date, use one of the following patterns:

.MMDDYY .MMYYDD .DDMMYY .DDYYMM
.YYMMDD .YYDDMM .TODAY .JULIAN

To validate a number, use one of the following patterns:

.NUMERIC .INTEGER .FLOAT

See [Chapter 3, “Advantage VISION:Transact Flags and Patterns”](#) for more information on validation patterns.

ED Operator - Edit Masks

If the Edit/Validation operator is ED, Operand B provides the edit mask. It consists of insertion characters and markers to indicate where the data characters are to appear in the edited field. The system delimiter serves as the marker character. Note if grouping characters are to be inserted into a numeric field, the EN operator should be used.

The edit pattern must be the same length as the screen field. An edited screen field cannot exceed 24 characters in length (the maximum length of Operand B). Leading and imbedded spaces are part of the edit pattern and are considered edit insertion characters.

The ED operation edits the source field for output and de-edits the input screen field.

When the source for an output field is a character field, editing is done as follows:

- The edit mask is placed on the screen.
- The first marker is replaced by the first source field character. The next marker is replaced by the next source field character, and so on.
- If there are more source field characters than markers, editing stops when the last marker is replaced; so the source field may be truncated on the right.
- If there are more markers than source field characters, the excess markers are replaced by spaces.

Examples of character editing are provided below. In these examples “b” represents a blank and “#” represents the system delimiter.

Source Field	Edit Pattern	Edited Field
123	##-#XYZ	12-3XYZ
12345	A##-###	A12-345
12345	##b##XX	12b34XX
b91285	##/##/##	b9/12/85
bbbbbb	##/##/##	bbbbbbbbb

When the source for an output field is numeric, it is converted to a character field and then right-justified to a field as long as the number of markers. Leading spaces are filled with zeroes. This character field is edited as above using the edit mask. If the source field has any decimal places, an error is printed during application generation. If the field is numeric, a maximum of 15 markers can be used.

When the source is invalid, missing or causes numeric overflow, editing is bypassed and an appropriate indicator character is placed in the screen field.

When the source field for output is empty (blank if character or zero if numeric), then the edit mask is not placed into the output display.

A screen field is de-edited only if it contains all its edit insertion characters and they are in the correct positions in the field. De-editing removes the edit insertion characters and compresses the remaining data characters to left-justify the field. This occurs before type 1 processing. An input screen field that contains only an edit mask and no data is blanked on input before type 1 processing.

Note: The ED operation is not performed if the output screen is recycled as input.

EC/ET Operators - Error Condition Codes

If the Edit/Validation operator is EC, this entry must contain one of the following three-character codes that identify an error condition. The conditions are classified as field or segment errors and are coded as Fnn or Snn, respectively.

Code	Error Condition
F01	Input is required but the screen field is blank.
F03	There are Sn validation statements with no MS operators and the screen field did not satisfy the validation.
F04	Numeric validation is specified (numeric specification on an S0 statement), but the screen field is not numeric.
F06	The result of the field action for this screen field contains invalid data. For example, the screen field value was added to the destination field and the resulting value was too big to fit in the destination field.
S01	The segment occurrence is not on the database.
S02	The segment occurrence is already on the data base and cannot be inserted.
S03	Fixed occurring segment cannot be inserted in a structured record; there is no empty slot into which the segment can be inserted. This error is not possible for DL/I or SQL databases.
S04	Variable length record exceeds maximum size for the file; insert or update cannot be performed. This error is not possible for DL/I or SQL databases.
S05	The screen field that has a segment action cannot be converted to the format (type, length, and scale) of the associated segment key.

If a segment has multiple key fields, only the first key field on the screen may have EC operators for segment actions.

If the Edit/Validation operator is ET, enter the error message text that is to override the preceding EC error.

Note:

If screen synchronized database fields are used for Edit/Validation, the Actions for those segments must appear in this screen segment or a higher level one (numerically lower).

Fields in the root segment of a direct dynamically synchronized database may be used for Edit/Validation if the synchronizing field is available.

Control Field Qualifier (43)

This entry only applies to Sn statements using the IF operator. In this case, it identifies the type of field used as the source in an IF comparison. Enter one of the following qualifiers:

Entry	Result
F	Flag field.
S	Screen field.

Control Field Name (44-51)

This entry only applies to Sn statements using the IF operator. In this case, it names the field used as the source in an IF comparison. If a screen field is used, it must be defined in the same screen segment or parent screen segment.

Control Operator (52-53)

This entry only applies to Sn statements using the IF operator. In this case, it specifies the way the control field and the control field value are compared. Enter one of the following compare operators:

Entry	Result
EQ	Control field equals the control value.
NE	Control field does not equal the control value.

Control Field Value Qualifier (54)

This entry only applies to Sn statements using the IF operator. In this case, it specifies a flag or constant to be used for the control field value. Enter one of the following qualifiers:

Entry	Result
C	Character constant.
D	Numeric constant.
F	Flag field.

Control Field Value (55-67)

This entry only applies to Sn statements using the IF operator. In this case, it names the flag field or specifies the constant to be compared with the control field.

Partial Field (68-72)

This entry indicates that only a part of a character string field is to be processed. Leave this entry blank to use the entire field.

Use this entry only when the Edit/Validation operator is used to validate input data, or with an MS or DF operator.

Starting Position (68-69)

This entry identifies the leftmost position of the field where the partial field is to begin. Enter a number from 1 to 99 that does not exceed the length of the entire character string field. Do not use a two-character partial field flag on this statement.

Number of Characters (70-71)

This entry indicates the length (number of characters) to be included in the partial field. Enter a number from 1 to 99. This number plus the starting position must not exceed the length of the entire character string field plus one.

Do not use a two-character partial field flag on this statement.

Operand A, B, or S (72)

This entry specifies the field to be partial fielded.

Do not enter A.

Enter B if Operand B is not a pattern and if this statement requires only part of Operand B.

Enter S if the Edit/Validation operator is relational (EQ, NE, LT, GT, LE, or GE) and this statement requires only part of the input screen field.

SS Statement - Structured Screen/COMMAREA Segment

In an application generation run or a definition run, SS statements are optional and are only required when a screen's contents are to be organized into a hierarchical structure of two or more screen segments.

The SS statement not only specifies the structure of the screen, but it also has a direct impact on the processing of the screen, especially in regard to looping. Note that the structure of a screen is a single path.

Screen Name (1-8)

Required entry

Enter the same screen name as on the SD statement.

Statement Type (9-10)

Required entry

Enter SS.

Screen Segment Number (11-12)

To indicate the relative order of this screen segment in the overall screen structure, enter a number from 01 to 99. Screen segment numbers need not be consecutive. The total number of screen segments must be less than ten.

A value of 1 is assigned if this entry is left blank.

A screen segment number must be unique within a screen definition.

Occurrences (14-17)

Enter the maximum number of occurrences (right-justified) of this screen segment within its parent. The root screen segment (segment number 1) must occur exactly once. If this entry is left blank, a 1 is assumed.

S0 Statement - Screen/COMMAREA Field

In an application generation run or a definition run, use an S0 statement to specify the content and processing of a screen field.

For an output screen field, the S0 statement specifies the source (database, temporary field, etc.) of the output data. There may be only one S0 statement specifying output for a single screen field.

For an input screen field, the S0 statement specifies the destination field for incoming screen data (database, temporary field, etc.). Closely associated with the destination is the action. The action determines whether the screen field is a key field used to locate, create, update, or delete a particular database segment occurrence, or a field used to update the data in that segment occurrence, or a field to be placed in a temporary field, flag field, or a working storage field.

The S0 statement may also:

- Specify automatic validation criteria for the screen field (additional lines of automatic validation are specified with Sn statements).
- Assign a name to the screen field.
- Assign a length to a virtual screen field.
- Specify a default value for the screen field.
- Specify descriptive text for fields when automatic formatting is used.

You may use multiple S0 statements for a single input screen field if the field has multiple destinations. The S0 attribute specifications - Input Required, Highlighting, etc. - for multiple S0 statements must be the same. The other S0 specifications - validation, default value, etc. - need not be the same.

A virtual screen field (V/S/A=V) is a field whose contents never physically appear on the screen; these fields are never sent to or returned from the terminal. On output, the virtual screen field is not displayed on the screen; however, it can be used to initiate loops on lower level screen segments that display unstructured data. On input, no data from the screen is automatically put in a virtual screen field; a request or procedure must load it. All S0 specifications, except positions [27-31], that are valid for a non-virtual screen field are valid for a virtual screen field.

If a screen layout is defined with SL statements, screen fields are indicated by a series of contiguous field designators. There must be an S0 statement for each field specified on the screen layout. If automatic formatting is requested, the screen fields are generated in the order of the S0 statements.

Screen Name (1-8)

Required entry

Enter the same screen name as specified on the SD statement.

S0 Statement Type (9-10)

Required entry

Enter S0.

Screen Segment Number (11-12)

This entry is optional if this field is in the root of the screen (segment 1). It identifies the screen segment that contains this field.

Indicate the screen segment that contains this field by entering a number from 01 to 99. Screen segment numbers need not be contiguous and the maximum number of screen segments is nine.

Virtual/Same/After (15)

Enter one of the following codes:

Entry	Result
Blank	<i>Next.</i> This S0 statement pertains to the <i>next</i> displayed screen field: assigned in left-to-right, top-to-bottom order.
V	<i>Virtual.</i> This S0 statement pertains to a virtual screen field which is not on the screen display; no field designators are used on SL statements for <i>virtual</i> screen fields. Just as virtual fields do not appear on the display, they also do not appear in the COMMAREA.
S	<i>Same.</i> This S0 statement applies to the same screen field (virtual or non-virtual) as the preceding S0 statement. This allows you to over define a screen field which is required for multiple input destinations or when the source field for output is different then the destination field for input.
A	<i>After.</i> This S0 statement pertains to a screen field that is displayed <i>after</i> (to the right of or below) all fields in dependent screen segments. <i>Subsequent S0 statements</i> in this screen segment will also pertain to screen fields displayed after their dependents. Only one field in a given screen segment should have this specification.

S0 statements specifying non-virtual screen fields are related to the field designators on the SL statements. You specify this positional relationship with a V/S/A entry of blank, S, or A. A blank (default) entry lets VISION:Transact relate S0 statements, in input order, to fields displayed at the terminal. An S entry lets you over define a displayed field with another S0 statement. An A entry lets you split a screen segment's display by placing screen fields after their dependents.

S0 statements specifying virtual screen fields are not related to the field designators on the SL statements. A V entry tells VISION:Transact not to assign this S0 statement to any displayed screen field.

Operand A, Source/Destination Field (16-24)

This entry consists of a qualifier and a source/destination field name. It is required for input fields with action codes other than N and required for output fields. Leave this entry blank if the Input/Output entry is P (IMS TM password).

Qualifier A (16)

This entry identifies the type of field or the file in which the field exists. Enter one of the following codes:

Entry	Result
Blank/N	Field from database or pseudo file whose File Type is FILE (as specified on an application generation RF statement).
1-9	Field from database or pseudo file whose File Type is FILE1-FILE9 (as specified on an application generation RF statement).
T	Temporary field.
F	Flag field.
W	Working storage field.
V	PCB mask field. Use only with fields specified as output only (position [25]=O).

Operand A Field Name (17-24)

This entry contains the field name which is to be the source for screen output and/or the destination (receiving field) for screen input. If this is a database, pseudo file, working storage, or PCB mask field, enter the field name exactly as it appears in the VISION:Transact file definition.

Input/Output (25)

Enter one of the following codes:

Entry	Result
Blank/O	Field is an output field only.
I	Field is an input field only.
B	Field is both an output field and an input field.
P	IMS: Field is an IMS TM password. (Use only in the root of the screen.)

Note:

1. Only one S0 statement per screen field may specify an Input/Output specification of O or B. IF O is specified, action and validation entries may not be specified.
2. The following rules apply to an IMS TM password field:
 - a. It must be in the root segment.
 - b. Only one password field per screen.
 - c. The following S0 statement may not refer to the previous S0 statement (it may not have V/S/A=S).
 - d. The password field is the default cursor position and is non-display. On this S0 statement, you should have only two specifications to the right of Screen Segment Number: Input/Output=P, and V/S/A=A or blank.
 - e. In a CICS screen definition run, an IMS TM password is treated as a protected field and will be blank.

Input Required? (26)

Use this entry with input or output/input (both) screen fields only.

If the Input/Output=P or O, leave this entry blank; otherwise, enter one of the following codes:

Entry	Result
Blank/N	An input value is optional for this screen field.
Y	An input value other than blank is required for this screen field.

When multiple input destination fields are specified for the screen field, specify the same value (Y or N) on each S0 statement for the screen field. Input is never required when an X action is in effect for an Sn set.

A default (DF operator) satisfies the Input Required test as long as the defaulted value is not blanks.

Highlighting (27)

This entry specifies the display intensity with which this field is to be displayed at the terminal. Use the same Highlighting on all S0 statements referencing the same screen field. Leave this entry blank for virtual screen fields or an IMS TM password, or enter one of the following codes:

Entry	Result
Blank/N	Normal intensity.
H	High intensity.
D	Dark intensity (does not display).

Cursor (28)

This entry specifies where the cursor is to be positioned when the screen is displayed. This cursor position may be specified for only one field of a screen.

Leave this entry blank for virtual screen fields or an IMS TM password.

If the cursor is to be positioned at the leftmost position of this screen field, enter Y. If you do not wish to explicitly position the cursor at this screen field, leave this entry blank or enter N.

If no screen field is specified for cursor positioning, the cursor is positioned according to the following priority:

1. IMS TM password.
2. Upper leftmost field on the screen used for input only.
3. First non-protected field used for both output and input.
4. Upper left corner of the screen (line 1, column 2).

Note: If the field is in a screen segment that is multiply occurring, the first occurrence of the field is used.

This entry must be the same for all S0 statements referencing the same screen field.

Error Handling (29)

This entry indicates whether the field is to be highlighted when the screen is re-displayed after an error is detected for the field.

Leave this entry blank or enter H to highlight this field in the event that an error is detected. Enter N to inhibit highlighting on error detection. This entry must be the same for all S0 statements referencing the same screen field.

Leave this entry blank for virtual screen fields, output-only fields, and an IMS TM password.

Protected? (30)

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

If this is an input or a both output/input field, enter Y to prevent modification of the value of the field as it appears at the screen.

When an input-only field is protected, the cursor stops at the field but the operator cannot enter data into it. When a both field (input/output) is protected, the cursor skips over the field.

This entry must be the same for all S0 statements referencing the same screen field.

This entry must be blank for virtual screen fields, output-only fields and an IMS TM password. (Output-only fields are automatically protected.)

Display Any Modification? (31)

When an input error occurs, this entry specifies whether the user sees the original value in the screen field, or if the user sees the value of the screen field after updates have occurred to it.

Enter one of the following codes:

Entry	Result
Blank/N	When the screen is re-displayed for any error, display the original contents of this screen field.
Y	When the screen is re-displayed for any error, display the modified contents of this screen field.

The contents of a screen field can be modified by a type 1, 2, or M request or procedure, a DF edit/validation operator defaulting over a blank screen field, or an ED edit/validation operator removing edit insertion characters.

Leave this entry blank for virtual screen fields, output-only screen fields, or an IMS TM password.

This entry must be the same on all statements referring to the same screen field.

Numeric (32)

This entry is used to specify numeric validation of input screen fields, where VISION:Transact:

- sets the numeric field attribute for the 3270 display so equipped and/or
- performs its own automatic numeric validation of the field.

Enter one of the following codes:

Entry	Result
Blank/N	No validation is done and the numeric attribute is not set.
Y	The numeric field attribute is set for the 3270 display station and automatic decimal, numeric validation is performed by VISION:Transact.
A	The numeric field attribute is set but no automatic numeric validation is performed by VISION:Transact.
V	The numeric field attribute is not set but automatic decimal numeric validation is performed by VISION:Transact.
E	The numeric field attribute is not set but floating point numeric validation is performed.

Leave this entry blank for an IMS TM password.

Note: If the Input/Output specification is O, this entry must be blank or N. For a virtual screen field, this entry must be blank, N, V, or E.

This entry must be the same on all statements referring to the same screen field.

Action (38)

This entry specifies the action which is to be performed with the data in the screen input field. These actions are grouped into two categories: segment actions and field actions. Segment actions that cause retrieval are performed between type 1 and type 2 requests or procedures; field actions between type 2 and type M requests or procedures.

Leave this entry blank for output only fields or an IMS TM password.

A field that is specified in the file definition of the database as a key (or search field for IMS databases) is eligible for use with a segment action. When multiple keys are specified for a database segment, all keys for that segment must have an action and all must use the same action. All segment actions for a database segment must be in the same screen segment.

[Figure 2-5](#) presents general rules for segment and field actions.

When dynamic selection of actions is specified for an Sn set (using IF operators), the action specified for the input screen field's S0 statement is used as the default action when all the IF relational tests are FALSE.

To process a segment using a key field, or possibly a search field (IMS database only), enter one of the following codes:

Entry	Result
I	<i>Insert</i> a segment whose key field is named as the destination field on this S0 statement, and whose value is provided by this screen field. If a segment already exists with this key field it is not inserted, a message is displayed and the interaction is terminated.
U	<i>Update</i> or replace the segment whose key field is named as the destination field on this S0 statement with the value provided by this screen field. If this segment does not exist, a message is displayed and the interaction is terminated.
V	This action code combines U and I. If the segment does not exist, it is inserted. If the segment already exists, it is updated.
M	<i>Match</i> and retrieve the segment with a key field equal to the value of this screen field. All field actions and validation specified for fields in the database segment as well as the keys are ignored.
D	<i>Delete</i> the segment whose key field is equal to the value of the screen field. If it does not exist, a message is displayed and the interaction is terminated. All field actions and validation specified for fields in the database segment as well as the keys are ignored. This action is not permitted on the root segment of an ISAM or VSAM ESDS file.

Entry	Result
Q	<i>Repeat</i> the action applied to the first key field in this database segment. If no previous actions were applied to this segment, repeat the action applied to this segment's parent. The Q action cannot be applied to the first key in the root segment.
X	<i>Ignore</i> automatic processing for fields in this data base segment as well as the keys. S0/Sn sets that name fields in this segment as a destination are not processed: validation, defaults, input required, or actions on those S0/Sn statements are not performed.

Note:

If a screen field is used as a control field to dynamically select a segment action, all validation is performed, regardless of the segment or field action associated with its destination.

The X action can only be specified as a dynamic segment action, i.e., it can only be specified when the Sn set contains an IF operator.

If all occurrences of a segment select the X action, none are screen synchronized and all occurrences of the segment under its screen synchronized parent are available for processing and output displays.

To process a non-key field in a segment having an I, U, or V action directed against it or inherited from its parent, use one of the field actions listed below. If a search field uses one of these actions, it is not considered a key field. Field actions are ignored and not performed on a segment whose action is M, D, or X.

Entry	Field Action
A	Add the value of the screen field to the destination field.
S	Subtract the value of the screen field from the destination field.
P	Unconditionally replace the value of the destination field with the value of the screen field.
R	Conditionally replace the value of the destination field with the value of the screen field. This is done if the screen field is not blank.
B	Blank or zero the destination field. If the destination field is a character type, set it to blank, otherwise set it to zero. In either case, the destination field is not modified if the screen field is blank.
N	Perform no automatic action with this screen field. Do not use this action with dynamically selected segment actions, only with dynamic field actions. Null can be selected dynamically to prevent the automatic updating of a field.

Note: Field actions for a given database segment must be specified in the same or lower level screen segment as the segment action.

1. Database and pseudo file segments being updated by field actions must specify at least one S0 statement with an updating segment action of I (insert), U (update), V (update or insert new occurrence of the database segment type), or Q (repeat, if the repeated action is update) for that segment occurrence.
2. If a non-root database or pseudo file segment specifies a segment action, then its parent must also specify a segment action.
3. The source/destination fields used with segment actions M, I, D, U, V, or X must reference a screen synchronized database or pseudo file. The specified field must be defined to VISION:Transact as a key/search field. If segment action I, D, U, or V, is specified, the database or pseudo file must be updatable. (Refer to the application generation RF statement in this guide.)
4. If multiple screen fields specify segment actions for the same database or pseudo file segment type (e.g., the segment has multiple key fields), all segment actions must be identical. (The Q action is useful in this case.)
5. In structured screens (see Chapter 6 in the *Advantage VISION:Transact Reference Guide*), if all input fields for a screen segment are empty (no data entered), and the Null Segment Processing entry on the SD statement is blank, N or B, the screen segment is considered to be missing and the corresponding actions are not performed.
6. A screen segment can contain field actions for a database field only if that screen segment or a parent screen segment contains a segment action for the database segment containing the field.

Figure 2-5 General Rules for Segment and Field Actions

Logic Level (39)

Leave this entry blank. It may only be used on an Sn statement.

Connector (40)

Leave this entry blank. It may only be used on an Sn statement.

Edit/Validation Operator (41-42)

This entry specifies operations to be performed on the data in the screen field. Except for the field descriptor, it is only valid with input screen fields (including virtual screen fields).

Enter one of the following codes:

Entry	Result
EQ	Input screen field is equal to Operand B or matches the pattern in Operand B.
NE	Input screen field is not equal to Operand B or does not match the pattern in Operand B.
LT	Input screen field is less than Operand B.
GT	Input screen field is greater than Operand B.
LE	Input screen field is less than or equal to Operand B.
GE	Input screen field is greater than or equal to Operand B.
DF	If the screen field is blank, operand B provides a default value.
DS	Operand B provides a character constant to describe this field when VISION:Transact provides an automatic format for the screen. See the SD statement for details on automatic formatting.
EC,ED, EN,ET	<i>Do not use</i> any of these operators on an S0 statement. These are Sn statement operators.
IF	<i>Do not use</i> this code on an S0 statement, it is only valid on an Sn statement. Any action specified on this statement serves as a default if no Sn "IF" comparisons are TRUE.

Entry	Result
LN	Operand B specifies the length of a virtual field, screen field, or COMMAREA field. If a destination is not specified for an input virtual field, LN is a required entry. If a destination is specified with no LN operation, the length of the virtual field is the same as the output edit length of the destination field. A virtual field is a character string field with a maximum length of 255 bytes. If an LN operation is specified for a non-virtual field and a corresponding field designator sequence is present on an SL statement, the LN operand value must match the length of the corresponding field designator sequence. If no LN operation is specified for a COMMAREA screen and there is no corresponding field designator sequence present on an SL statement, the length of the COMMAREA field is the same as the output edit length of the Source/Destination Field.
MS	<i>Do not use</i> this code on an S0 statement, it is only valid on an Sn statement.
NC	Operand B specifies an alternate name for this input screen field that can be up to 24 characters long. When specified, this name will be used in the COBOL statements that define the COMMAREA. If no NC operator is provided for a field, the NM operand is used. If neither an NC or an NM operator is provided, the Source/Destination Field name for the screen field is used as the COBOL element name. The NC operator can be specified even when the NM operator is also specified.
NM	Operand B specifies a name for this input screen field (uniqueness is not verified). If an NM operator is not used for a screen field, the field takes its name from the destination field. If duplicate names are assigned by NM operators for a screen input field, an error is issued.
SE	<i>Do not use</i> this code on an S0 statement, it is only valid on an Sn statement.

Operand B (43-67)

Operand B serves many purposes. For Edit/Validation operators, it specifies the field name, constant, or pattern for a relational test; or it specifies screen field names, field lengths, default values, or descriptions for automatically formatted fields.

Leave this entry blank for an IMS TM password.

Qualifier B (43)

Leave this entry blank when the Edit/Validation operator is DS, LN, or NM. Code an S entry for the SE operator.

Identify the source of the field used for edit/validation by entering one of the following codes:

Entry	Result
Blank/N	Field from file or pseudo file whose type is FILE.
1-9	Field from file or pseudo file whose type is FILE1-FILE9.
S	Another field in the same screen segment or a higher level one (numerically lower).
T	Temporary field.
F	Flag field.
W	Working storage field.
C	Character constant.
D	Numeric constant.
P	Pattern for validation (EQ or NE operators only).

Edit/Validation Field (44-67)

This entry identifies the operand field name or constant used for edit/validation operators. If the operator is DS, Operand B must be a character constant.

If the edit/validation operand for this statement is:

Operator	Operand B
EQ,NE	Enter a constant, field name, or a validation pattern to validate the input screen field. See below for details on pattern validation.
LT,GT, LE,GE	Enter a constant or field name to validate the input screen field.
DF	Enter the default value to be assigned to this screen field.
DS	Enter the descriptive text to be displayed with the contents of this screen field.

Operator	Operand B
LN	Enter the length of the virtual screen field specified by this statement. Maximum length is 255 characters.
NM	Enter the unique name to be assigned to this screen field.

Note:

1. If screen synchronized database fields are used for edit/validation, the Actions for those segments must appear in this screen segment or a higher level one (numerically lower).
2. Fields in the root segment of a direct synchronized database may be used for edit/validation if the synchronizing field is available.

For pattern validation, the edit/validation operator must be EQ or NE. Enter a P in Operand B qualifier [43], a period (.) in position [44], followed by the pattern name.

To validate a date, use one of the following patterns:

.MMDDYY .MMYYDD .DDMMYY .DDYYMM .YYMMDD .YYDDMM
.TODAY .JULIAN

To validate a number, use one of the following patterns:

.NUMERIC .INTEGER .FLOAT

See Chapter 3 for more information on validation patterns.

Partial Field (68-72)

This entry indicates that only a part of a character string field is to be processed. Leave this entry blank to use the entire field.

Use this entry only when the Edit/Validation operator is used to validate input data, or with a DF operator.

Starting Position (68-69)

This entry identifies the leftmost position of the field where the partial field is to begin. Enter a number from 1 to 99 that does not exceed the length of the entire character string field. Do not use a two-character partial field flag on this statement.

Number of Characters (70-71)

This entry indicates the length (number of characters) to be included in the partial field. Enter a number from 1 to 99. This number plus the starting position must not exceed the length of the entire character string field plus one.

Do not use a two-character partial field flag on this statement.

Operand A, B, or S (72)

This entry specifies the field to be partial fielded.

Leave this entry blank or enter A to apply the partial fielding to a character field in Operand A (source/destination field). The segment action must not be M, U, D, I, V, Q, or X; or the field action must not be A or S. If this statement defines an output screen field, then part of Operand A is placed in the screen field starting at the left. If this statement defines an input screen field, then the entire screen field is applied (depending on the action code) to part of Operand A.

Enter B if the partial field applies to the edit/validation, or default operator (DF). This entry is valid only with validation and default operators. It is invalid if Operand B is a pattern.

Important! Do not enter S. It is only valid on an Sn statement.

TF Statement - Temporary Field

In an application generation run or a definition run, use TF statements in a request to explicitly define fields used for temporary results by the application program.

If TFREQ=Y is specified in M5PARM, then all temporary fields must be explicitly defined on a TF statement. If TFREQ=N is specified in M5PARM, then implicit temporary field definitions are possible; they do not require a TF statement. For more information about implicit temporary fields, see Chapter 9 of the *Advantage VISION:Transact Reference Guide*.

Request Name (1-8)

Required entry

Enter the name of the request as it appears on the ER statement.

Statement Type (9-10)

Required entry

Enter TF.

Field Name (11-18)

Required entry

This entry names the field being defined. Temporary field names must be unique across an entire application.

Field Length (19-21)

This entry specifies the length of a field. Enter the field length within the following ranges:

Field Type Code	Field Length	Field Length Default
Blank/C	1-255 bytes	16 bytes
Z	1- 15 bytes	15bytes
P	1- 8 bytes	8 bytes
E	4 bytes only	4 bytes
F	1- 4 bytes	4 bytes

Type F Value Ranges:

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

Field Type (22)

Enter one of the following codes:

Entry	Result
Blank/C	Character string.
Z	Zoned decimal number.
P	Packed decimal number.
F	Fixed point decimal number (2's compliment form).
E	Floating point decimal number.

Decimal Places (23)

This entry specifies the number of implied decimal places in a numeric field. This number must not exceed the maximum number of digits assigned for the field length. Enter a number from 1 to 9. Leave this entry blank and no decimal places are assigned.

Output Edit Length (27-28)

This entry specifies the size of field that is used to display this field. Leave this entry blank and the maximum number of positions to display this field is assumed. Otherwise, enter a smaller number to restrict the data included in this field.

Initial Value (29-44)

This entry is the value used to initialize the temporary field. Enter any character string or number. A number may be preceded by a sign or include a decimal point. Leave this entry blank and the field is initialized to zero if it is numeric or to blanks if it is a character field.

A temporary field is initialized to this value at the beginning of each interaction, unless this is a recycle interaction or CICS conversational interaction.

Comments (45-72)

Provide documentation for this temporary field. Enter any helpful information.

UC Statement - Utility Control

The UC statement is only input during a library utility run (M5UTIL). It allows you to establish and maintain your VISION:Transact library. With it, you can perform the following operations:

- Initialize libraries.
- Create a backup copy of a library.
- Restore 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.

Note:

1. The UC statement is used in conjunction with the M5UTIL program.
2. Comment (AA) statements are the only other valid statement type in the run. They must appear after the first UC statement.
3. Multiple UC statements may be included in a single M5UTIL job step.

Run Name (1-8)

This optional entry is ignored.

Statement Type (9-10)

Required entry

Enter UC.

Operation Code (11-14)

Required entry

Enter one of the following library operations:

Entry	Result
INIT	Initialize a VISION:Transact library.
DUMP	Create a backup copy of a library.
REST	Restore a library from a backup copy.
COND	Condense or reorganize a library.
COPY	Copy definitions from M5LIBn (n=1-9) to M5LIB.
MERG	Copy the entire contents of M5LIBn (n=1-9) to M5LIB.

Generic Item Name? (15)

This entry is used with the COPY operation. Enter one of the following codes:

Entry	Result
Blank	Item name entered on this UC statement is the full item name.
Y	Item name entered on this UC statement is a generic or partial name. Any item having this partial name in its name is copied.

Item Type Code (16-17)

This entry is used with the COPY operation and it identifies the type of item to be processed. Enter one of the following codes:

Entry	Result
Blank	All item types.
FD	File definitions only.
RQ	Requests only.
RG	Request groups only.
SD	Screen definitions only.

Item Name (18-25)

Enter, left-justified, the full or generic name of the item to be processed by the COPY operation.

Library DD/DTF Name (34-41)

Identify the library to be processed.

Entry	Result
Blank	Default library to be processed by the operation specified on this UC statement: INIT (Initialize M5LIB); DUMP (Dump M5LIB); REST (Restore M5LIB); COND (Condense/ Reorganize M5LIB); COPY (Copy from M5LIB1 to M5LIB); MERG (Merge entire contents of M5LIB1 into M5LIB).
M5LIBn	Identifies a particular M5LIBn (n=1-9) to be processed.
Note: Items are always copied or merged into M5LIB.	

Replace Option? (42)

With a COPY or MERG operation, enter one of the following codes:

Entry	Result
Blank/N	Suppress the COPY or MERG operation if the item exists on M5LIB.
Y	Replace items on M5LIB with like-named items from the sending library.

Replace Rule (43-45)

Identify the type of replacement to take place when the COPY or MERG operation is used and a like-named item exists on M5LIB.

Entry	Result
Blank	Replace an item on M5LIB with a like-named item in another library.
OLD	Replace an item on M5LIB with an older (not as recently updated) like-named item from the sending library.
NEW	Replace an item on M5LIB with a newer (more recently updated) like-named item from the sending library.

Directory Blocking Factor (46-49)

Valid only with the INIT, COND, and REST operations.

Enter the number of entries in a library directory block, right justified (BDAM only).

Entry	Result
Blank	Specification from MARKLIBP is used.
1-9999	Number of entries in a library directory block.

Note:

1. If MARKLIBP directory blocking factor is zero, VISION:Transact 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.
2. 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.
3. The limits VISION:Transact uses for each supported device are provided in the description of MARKLIBP in the *COMLIB Installation Procedures* guide.

Purge Expired Items? (50)

Valid only with the REST operation.

Entry	Result
N or blank	Items with expired dates are restored along with all other items during a restore operation.
Y	Purges all items that have expired dates when a restore operation is performed.

Expiration date for an item is specified using the IT statement.

Advantage VISION:Transact Flags and Patterns

This chapter describes the available VISION:Transact flag field and validation patterns.

VISION:Transact Flags

VISION:Transact provides flags or internal indicators which you can interrogate or set to indicate the existence of certain conditions. Flag fields are identified by the qualifier F. Their general uses are shown in [Figure 3-1](#). Flags are not re-initialized during a recycle interaction, except for the date and time flag fields.

Flag Name	Application
TODAY DATE ISDATE ¹ JULIAN	Provides the operating system date (in one of four possible formats) to the user. Note: ISDATE = International Standard Organization (ISO) date format.
TIME	Provides the time of day that execution was started.
CSTATUS RETURNCD	Provides status information regarding CALL or DLI PR statements.
ATTNID	Provides information about the terminal program function keys that caused transmission of the input message.
ERROR	Provides an automatic count of the number of times the SE (Set Error) operator is executed.
TRANCODE	Provides the name of the current transaction that is processing the input screen data.

Figure 3-1 VISION:Transact Flags (Page 1 of 2)

Flag Name	Application
PROGTYP	Indicates whether a CICS program is running as a 3270 Terminal program or a COMMAREA program.
SYNCOK	Indicates whether a CICS SYNCPOINT is allowed for a COMMAREA application.
GOSC	Provides the type of the request (1, 2, or M) from which a GO/NS/GS SCREEN CONTROL branch was issued.
TERMID USERID OPERID OPERCL (CICS only)	Provides information about the terminal or operator.
MSGLINE	Enables you to explicitly place the error message line on the screen, and to place user data in the error message line on non-error displays.
PSTART or PS PNUMBER or PN RSTART or RS RNUMBER or RN MSTART or MS MNUMBER or MN LSTART or LS LNUMBER or LN	Provides numeric fields to be used as counters or indexes (useful during back branch operations). Abbreviations (two characters) allowed in partial fielding entries on PR and SC statements.
HIGHVALU LOWVALU	Provides high value or low value character strings useful in the testing and setting of high or low values.
PAGENO	Used to display page numbers for output screen pages.
SCRNAMEI SCRNAMEO	Provides the name of the input and output screen being processed in the interaction.
TCONTROL TAPPFLOW TPROC TSEGACT TILOOP TDLOOP TSSDBIO TDSDBIO	Provides the ability to enable or disable all or individual execution tracing options, provided the application was generated to include tracing on the ?T statement.
INVFIELD	Provides a count of the number of times a field was set invalid (overflow, divide check, bad packed decimal data, and so on).

Figure 3-1 VISION:Transact Flags (Page 2 of 2)

When specifying a flag name, use F in the qualifier column.

Today, Date, ISDATE, and Julian Flags

All four flags produce the same date but in the following different formats:

TODAY	Six characters as follows: MMDDYY (where MM=month, DD=day, YY=year), e.g., 101493. This format can be controlled by M5PARM. The format shown is the default.
DATE	Twelve characters as follows: Three alphabetic characters for month, space, two numerals for day, comma, space, and four numerals for year. For example, OCT 14, 1993. This format can be controlled by M5PARM. The format shown is the default.
ISDATE	Eight characters as follows: YYYYMMDD (where YYYY=year, MM=month, DD=day), e.g., 19931014. Note: ISDATE = International Standard Organization (ISO) date format.
JULIAN	Five characters as follows: YYDDD (where YY=year, DDD=day), e.g., 93288.

The formats for the TODAY and DATE flags are influenced by specification in M5PARM (refer to the *Advantage VISION:Transact for CICS Installation Guide*).

Time Flag

The TIME flag contains the time of day (on a 24-hour clock) when application execution began. The data is stored as an eight-byte character string and takes the form:

HH.MM.SS

where:

HH=hours MM=minutes SS=seconds

Single digits will contain a leading zero. For example, six minutes and four seconds after 9:00 A.M. will be represented as 09.06.04, and 12 seconds and 46 minutes after 4:00 P.M. will be represented as 16.46.12.

CSTATUS and RETURNCD Flags

The status of the Call (using either keyword, CALL or DLI) in IMS TM applications is determined from two flag specifications: RETURNCD and CSTATUS. In CICS applications, you can use only the CSTATUS flag.

Call status information is obtained from the Call Status (CSTATUS) flag for IMS TM and CICS. The CSTATUS provides a method of determining in a VISION:Transact request if a call was suppressed by VISION:Transact. A call will be suppressed if a parameter specifies an invalid or missing field.

The CSTATUS is a four-byte character string flag field which may have the following values:

PMIS	Parameter data field missing.
PINV	Parameter data field invalid.
(blank)	Call successful.

The CSTATUS flag should be examined immediately after a call, since a suppressed call would not allow the called routine to be executed.

The RETURNCD flag is a VISION:Transact flag that shows the result of the call as set by the called routine (IMS TM only).

The VISION:Transact Return Code flag field (RETURNCD) values are set by the called routine, while the Call Status flag field (CSTATUS) values are set by VISION:Transact. Following any call, the contents of RETURNCD may be examined (after CSTATUS flag) to determine the results of the call as set by the called routine. The value in the RETURNCD field is the value in General Register 15 upon return to VISION:Transact from the called routine. This value may be set by an Assembler program by simply loading the register with the desired value; COBOL programs use the RETURN-CODE keyword.

The RETURNCD flag is a four-byte, fixed point field. However, VISION:Transact edits it as though it were a two-byte fixed point field, so its display width is seven and the maximum printable value is 99,999.

When testing for a specific value in the RETURNCD, the developer must know the possible setting, which depends on the called routine.

ATTNID Flag

The ATTNID flag allows you to determine which terminal key was used on the last input.

The ATTNID flag is a five-byte character field which may contain one of the following values:

ENTER	Operator pressed the Enter key.
PFnnn	Operator pressed PF key nnn where nnn may contain a value from 001 to 036.
CLEAR	CICS: Operator pressed the Clear key. Clear is only passed to a VISION:Transact application if the CICSKEY handling entry in M5PARM or the RC statement indicates that the Clear key is to be processed.
PAnnn	CICS: Operator pressed PA key nnn where nnn is a value of 001 to 003. PA keys are only passed to a VISION:Transact generated application if the CICSKEY handling entry in M5PARM or the RC statement indicates PA keys are to be processed.

Note:

1. If CICS BMS paging commands have been assigned to PA or PF keys, and multiple pages of a screen defined with Output Control entry of A in the screen definition or OC OUTLIM in screen control have been output, then the PF or PA keys will not be passed to the application while the operator is paging, but are intercepted by BMS.
2. Since the ATTNID flag is a character field, PF or PA key numbers must always be specified with leading zeroes (e.g., PF1 is not the same as PF001).

ERROR Flag

The ERROR flag is used in conjunction with the SE (Set Error) operator in type 1, 2, and M requests. The ERROR flag is a two-byte fixed point field which is initialized to zero at the start of each interaction. It is automatically incremented by one each time an SE operator is executed.

The value of this flag can be interrogated to determine whether input screen fields are in error. It can also be displayed in the text of an MS operator. This technique provides a means for indicating the number of screen fields with errors.

TRANCODE Flag

The TRANCODE flag is an eight-byte field. On entry to a VISION:Transact generated application, the current transaction name is placed in this flag field. For IMS TM conversational transactions, the TRANCODE flag is initialized to the transaction code contained in the SPA.

You can manipulate this flag by permitting the user to enter a new value on the screen or by supplying one yourself in request logic. The transaction specified in the flag processes the next screen's input. The contents of this flag may be modified by the application at any time prior to the building of the output screen.

The TRANCODE flag facilitates the sharing of screens between applications, because you are not required to specify in the screen processing definition a specific transaction to process a specific screen. The value in the field determines the transaction that will process the screen entered. Thus, a screen can be input to various transactions.

IMS TM: The use of the flag in an IMS TM application requires that it appear on the screen and have an S0 statement associated with it. If you want, the field can be specified as dark and protected to guard against user error and allow only you to update the flag procedurally in a request. Note that the flag field must not appear in a multiply occurring screen segment and must have a B (both) Input/Output specification. The value in the field is used to determine the transaction that will process the output screen on subsequent input processing.

CICS: To use the flag in a CICS application, you need not place the TRANCODE flag on the screen. Instead, you can simply procedurally update the flag in a request. To update the flag, place the name of the next transaction in it (CICS transaction name is limited to four characters). This update can be done any time before the application completes building the output screen. The VISION:Transact generated application will use the flag as the TRANSID upon RETURN to CICS. See the *Advantage VISION:Transact for CICS Environment Guide* for more information on using the TRANCODE flag.

PROGTYP Flag

The PROGTYP flag is a one-byte character flag that indicates whether a VISION:Transact CICS program is running as a 3270 Terminal program or a COMMAREA program. When running as a 3270 Terminal program the flag contains a T. When running as a COMMAREA program, the flag contains a C.

SYNCOK Flag

The SYNCOK flag is a one-byte character flag that indicates whether a CICS SYNCPOINT is allowed for a CICS COMMAREA program. The flag contains a Y if the VISION:Transact CICS COMMAREA program was initiated via a Distributed Program Link (DPL) with a SYNCONRETURN option. The flag contains an N if the program was initiated through DPL without the SYNCONRETURN option.

GOSC Flag

The GOSC flag is a one-byte character field that is used together with the request branching location SCREEN CONTROL. When a type 1, 2, or M request executes a GO/GS/NS SCREEN CONTROL branch, then this field will contain the type of the request that executed the branch. If no such branch is issued, then GOSC contains a blank. The only possible values for GOSC are 1, 2, M, or blank.

TERMID, USERID, OPERID, and OPERCL Flags

These flag fields contain the identification of the terminal, the user, or operator who initiated the transaction, and, for CICS applications, the operator class.

TERMID is an eight-byte character field. In an IMS TM application, TERMID is initialized to the LTERM name in the I/O PCB. In a CICS application, it is initialized to the value of EIBTRMID in the Exec Interface Block. Only the leftmost four bytes of TERMID are significant in CICS. The rightmost four bytes contain blanks.

The USERID flag is an eight-byte character string field. Under CICS, the flag contains the user identifier of the person who is signed on; it contains blanks if the CICS Release is prior to 1.7 or if the VISION:Transact task is not initiated from a terminal. Under IMS TM, the USERID flag contains the user identification field from the I/O PCB; this information is also available through the OPERID flag.

OPERID is an eight-byte character field. In an IMS TM application, OPERID is initialized to the USERID in the I/O PCB. In a CICS application, the leftmost three bytes of OPERID are initialized to the value returned by the CICS ASSIGN OPID command. Bytes four through eight are blank.

CICS: OPERCL is a three-byte character field. In a CICS application, OPERCL is initialized to the value returned by the CICS ASSIGN OPCLASS command. For IMS TM applications, this flag field contains blanks.

MSGLINE FLAG

When your application encounters an error condition, it re-displays the screen that was input to the application with an error message. This error message by default is placed on the bottom line of the terminal.

The MSGLINE flag is 255-byte character field and it is initialized to blanks at the start of each interaction. It allows you to place the line on which the error message will be displayed anywhere in the root screen segment. In addition, because the contents of the MSGLINE flag are displayed during normal output processing, you can place data to be displayed into the MSGLINE flag field.

If you do not reference the MSGLINE field on an S0 statement, then the message line is automatically placed on the last line of the screen. To place the message line somewhere other than the bottom line of the screen, you first enter a series of at least 79 screen field designators on an SL statement. This screen field defines the location of the error message when the input screen is redisplayed. You then write an S0 statement (in the root screen segment) for this screen field that names the flag field MSGLINE as its source. This screen field must not be a virtual screen field. When the application enters an error redisplay cycle, the message is put into this screen field.

To output data into the error message line, move whatever data you want into the flag MSGLINE in a request. That data will be displayed in the error message line when the screen is output as long as the MSGLINE flag is an output field. This data will not be displayed during an error cycle unless flag MSGLINE was named as Operand B of the MS operator.

Dynamic Partial Field Flags

There are eight flag fields that can be used in partial field operations. They are PSTART, PNUMBER, LSTART, LNUMBER, MSTART, MNUMBER, RSTART, and RNUMBER. The field names can be abbreviated to PS, PN, LS, LN, MS, MN, RS, and RN, respectively. These fields are four-byte, fixed point, integer fields that are initialized to zero at the start of an interaction.

The two-character abbreviations are used in the Partial Field Start and Partial Field Length specifications on PR and SC statements. In addition, the field names or abbreviations may be used as A, B, or C Operands on PR and SC statements, A or B Operands on S0 statements, or B Operands on Sn statements. Dynamic partial field flags cannot be used in the Partial Field Start or Partial Field length entries of S0 or Sn statements.

For example, if the value of LS is 10 and value of RN is 3, and LS is placed in Partial Field Start and RN is placed Partial Field Number, then positions 10, 11, and 12 of the referenced field will participate in the operation. Note that this is equivalent to placing a 10 in Partial Field Start and a 3 in Partial Field Number.

When you use these dynamic partial field flags, you must be aware of the rules that apply to partial fielding:

- Partial field start must be between 1 and the length of the field.
- Partial field length must be greater than zero.
- Partial field start plus partial field length must not exceed the length of the field.
- Neither partial field start nor partial field length may be invalid.

If these rules are violated, then one of the following occurs:

- If the field being partial fielded is a source field for some operation, then the field is treated as an invalid field.
- If the field being partial fielded is a result field, then the entire field becomes invalid.

HIGHVALU and LOWVALU Flags

The HIGHVALU flag is an eight-byte character field that contains high values (X'FF's). The LOWVALU flag is an eight-byte character field that contains low values (X'00's). These flags can be useful in the testing and setting of high or low values in other fields.

PAGENO Flag

The PAGENO flag is a two-byte fixed point binary field that contains the page number of the output screen. When page numbers are desired on output screen pages, a screen field must be defined on the screen layout and the corresponding S0 statement for the screen field must name flag PAGENO as the source field for output.

If flag PAGENO is interrogated during output processing, it is important to understand that the contents of the flag reflects the page number of the previous branch to output executed by the screen's type O request. The flag contains a 1 on entry to a screen's type O request and is incremented by 1 *after* a screen page is output by VISION:Transact.

SCRNAMEI and SCRNAMEO Flags

These flags are eight-byte character string fields.

The SCRNAMEI flag contains the name of the screen or COMMAREA section that was input to the interaction. During a recycle interaction (the output screen is recycled as input), SCRNAMEI contains the name of the recycled screen upon input. Flag SCRNAMEI is initialized at the start of each interaction cycle to the name of the input screen; this includes recycle and CICS conversational interactions.

The SCRNAMEO flag contains the name of the screen or COMMAREA section currently being output in the interaction. This flag contains blanks during input processing (type 1, 2, or M requests or subroutine requests called from these request types). During output processing, this flag contains the name of the output screen currently being processed. Flag SCRNAMEO is initialized at the start of each interaction cycle to blanks; this includes recycle and CICS conversational interactions. The flag is then re-initialized during screen control for each output screen selected. The name of the selected output screen is placed in the flag prior to execution of the type O request and building of the output screen.

Execution Tracing Flags

There are eight flag fields that can be used to enable or disable all or individual execution tracing options, provided an application has been generated to include tracing. These trace flags which correspond to the entries on the ?T statement, are one-byte character fields and have the following meanings and allowable settings:

TCONTROL	Trace Control
	0 indicates that all tracing is disabled, 1 indicates tracing is enabled and that individual traces are enabled or disabled, based on the contents of their associated trace flag.
TAPFLOW	Application Flow Tracing
	0 indicates disabled, 1 indicates level 1 tracing is enabled, and 2 indicates level 2 tracing is enabled.
	Level 1 outputs all level 1 application flow messages (AFnn) and teleprocessing interface calls for screens and saved working storage/pseudo files (TPnn messages).
	Level 2 outputs all level 1 and 2 application flow messages and all TP interface calls and I/O areas for screens and saved working storage/pseudo files.
TPROC	Procedure Tracing
	0 indicates disabled and 1 indicates enabled which also enables trace commands/operators.
TSEGACT	Segment Action Tracing
	0 indicates disabled and 1 indicates enabled.
TILOOP	Intermediate Structure Loop Tracing
	0 indicates disabled and 1 indicates enabled.
TDLOOP	DSDB Loop Tracing
	0 indicates disabled and 1 indicates enabled.

TSSDBIO	SSDB I/O Call Tracing
	0 indicates disabled, 1 indicates level 1 enabled (I/O areas are not displayed), 2 indicates level 2 enabled (I/O areas are displayed).
TDSD BIO	DSDB I/O Call Tracing Level
	0 indicates disabled, 1 indicates level 1 enabled (I/O areas are not displayed), 2 indicates level 2 enabled (I/O areas are displayed).

The TCONTROL flag is initialized to 1 at the beginning of application execution if the ?T statement contains an entry of 1 in Trace Control; otherwise it is initialized to 0. The remaining flags are initialized at the beginning of application execution to the value of their corresponding ?T statement entry. Note that a value of blank on the ?T statement is treated as 0 in the corresponding flag field.

The trace flags can be modified by requests or procedures as well as screen field actions in order to enable or disable all or individual tracing options. Unlike other flags, trace flags retain their value between interactions; they are not re-initialized until application termination which can span multiple interactions. For IMS TM, application termination occurs after all input messages have been processed for a given transaction. For CICS TS, application termination occurs when the pseudo conversation has been terminated, which can span multiple transactions.

Note that if a ?T statement is not included during APPGEN or the ?T statement contains an entry of P in Trace Control, then tracing is not included in an application and any modifications to the trace flags are ignored. The trace flags cannot be used to enable or disable tracing unless a ?T statement has been included during APPGEN with a Trace Control entry of 0 or 1.

Modifications to the trace flags do not affect the tracing options being used until the request or procedure terminates. The new tracing options will become effective upon exit from the request or procedure and remain effective until they are changed or the application terminates. In the case of field actions modifying the flags, the trace options do not become effective until in-storage update processing has been completed, thus becoming effective for the first type M request or procedure. The reason the new trace options do not become effective until execution of a request or procedure completes is to avoid incomplete trace messages being output. For example, if the trace flags were modified after a database loop had been established within the request or procedure, incomplete and misleading looping and I/O call trace messages could be output.

If it is desired to change the tracing options in effect during the execution of a given request or procedure, then a subroutine request or procedure could be used to modify the appropriate flags. The new tracing options would become effective upon exit from the subroutine and return to the calling request or procedure.

When application execution begins, the trace options in effect are those specified on the ?T statement. Type 1 processing is the earliest point within the interaction cycle for which the trace options in effect can be modified. Therefore, the only way to modify the trace options in effect prior to type 1 processing (for example, trace messages AF01-AF14), is to modify the trace flags in a previous interaction.

INVFIELD Flag

The INVFIELD flag is a two-byte fixed point binary field that contains a count of the number of fields that were set invalid during an interaction. A field is set invalid when it is a result field of an arithmetic or assignment operation that fails due to overflow, divide check, bad packed decimal data, record or segment data does not exist, and so on. A non-zero value can imply that the program logic should follow some corrective or remedial action path during termination.

Validation Patterns

A pattern, identified by the qualifier P, may be used as the B Operand of one of the relational operators EQ or NE. A pattern may be used on S0, Sn, SC and PR statements. A pattern is used to describe the contents of the character field named in Operand A of the relational expression on a PR or SC statement. If the pattern describes Operand A, the relationship is considered to be equal and the result of the relational operand (EQ or NE) is evaluated accordingly. For S0 and Sn statements, a pattern is used to describe the contents of a screen field. If the pattern describes the screen field, the relationship is considered equal and the result of the relational operand (EQ or NE) is evaluated accordingly.

Named Patterns

You can use several named patterns that are predefined by VISION:Transact. To use a named pattern, enter a period(.) in the first position of Operand B followed by the pattern name starting in position two.

Pattern names and their values are as follows:

Pattern	Value
MMDDYY MMYYDD DDMMYY DDYYMM YYDDMM YYMMDD	These patterns describe the various six-character date formats. All of these require six left-justified numeric digits. This validation will assure that year (YY) is numeric, month (MM) is numeric and in the range 01 through 12, and day (DD) is numeric and is valid within month (includes validation for leap years).
TODAY	This pattern corresponds to the format of the TODAY flag field. It will be the same as one of the formats above. TODAY flag format can be specified in M5PARM and this can be installation specific. See the description of the TODAY flag earlier in this chapter.
JULIAN	This pattern describes a date in the format YYDDD, where year (YY) must be numeric and day (DDD) must be numeric in the range 001-365 (001-366 for leap years).
NUMERIC	This pattern describes a field that, except for leading and trailing spaces, contains at most one sign (a minus, "-"), numeric digits, grouping characters, and at most one decimal point (.). If the field contains a sign, then it must immediately precede the first numeric digit or the decimal point. If field has a decimal point, it must be followed by at least one numeric digit. A blank field is considered to be numeric. The field may contain grouping characters (for example, a comma); however, the grouping characters must separate groups of three digits (except for the leftmost group which may contain one or two digits).
INTEGER	This pattern describes a field that, except for leading and trailing spaces, contains at most one sign (a minus, "-"), numeric digits and grouping characters. If the field contains a sign, it must immediately precede the first numeric digit and there must be at least one digit. The field may not contain a decimal point. A blank field is considered to be an integer. The field may contain grouping characters (for example, a comma); however, the grouping characters must separate groups of three digits (except for the leftmost group which may contain one or two digits).
FLOAT	This pattern describes a number in floating point (scientific) notation. A field is considered to be in floating point format if it is a decimal number followed by an exponent. An exponent is the character E followed by a sign, + or -, and a one or two digit number. The sign may be omitted, in which case a plus (+) is assumed.

Index

Symbols

?C operator, 2-35
?T statement, 1-10, 2-1, 2-35
?X operator, 2-35

Numerics

3270 terminal, 2-79
3270 Terminal program, 3-2, 3-8
3270 Terminal programs, 1-1

A

AA statements, 1-7, 1-14
alias, 2-31
ALIAS entry, 2-25
AND logical connector, 2-33
AND relationship, 2-19
AND/OR condition, 2-19
APAR TR00115, 2-57
APPGEN, 1-8, 1-11, 1-12, 1-13, 2-5, 2-32
 JCL, 1-9
 run, 2-9, 2-13, 2-74
APPGEN run, 1-5, 2-47, 2-60
APPLCTN statement, 1-9
application flow tracing level, 2-3
application generation run (APPGEN), 1-8

arithmetic operator, 2-35
ATTNID flag, 3-6
authorization ID, 2-24, 2-59

B

back branch limit, 2-11
BIND, 2-24
BMS source statements, 1-5, 1-11, 2-77
BMS statements, 1-4
branch operator, 2-35
buffer size, 2-14

C

Call PR statement, 2-43
catalog listing/retrieval, 2-7
catalog maintenance, 1-3
catalog request, 2-6
catalog retrieval, 1-14
catalog retrieval statements, 1-12
cataloged request operation, 1-7
character editing, 2-91
Character string data, 2-21
CICS, 2-2
 ASSIGN OPCLASS command, 3-9
 Program Control Table (PCT), 2-77

CICS conversational interaction, 2-114
CICS screen definition run, 2-100
CICS SYNCPOINT, 3-2
CICS transaction code, 2-46
CICS TS, 1-9, 2-11, 2-27, 2-29, 2-51
CICS TS environment, 1-1
CICS TWA, 2-45
CICSKEY
 handling, 2-50
CICSKEY specification, 2-50
Clear key, 2-50
COBOL statements, 2-88
Color attributes, 2-80
COMMAREA, 2-45
 section name, 2-62
COMMAREA control statements, 1-10
COMMAREA Definition, 2-73
COMMAREA Definition Statements, 1-11
COMMAREA definition statements, 1-5
COMMAREA program, 3-2, 3-8
COMMAREA programs, 1-1
comments
 AA, 1-7, 1-13, 1-15
COND, 1-14
control field name, 2-93
control field qualifier, 2-93
control field value, 2-94
control field value qualifier, 2-94
control operator, 2-93
CONV parameter, 2-11
conversational interaction, 2-114
COPY, 1-14
count fields, 2-30
CR statement, 1-8, 2-6
CR statements, 1-12

creator ID, 2-24
CSECT name, 2-44
CSTATUS flag, 3-4
CT operations, 1-3
CT statement, 1-6, 2-7, 2-8
CT statements, 1-14

D

data entry database (DEDB), 2-23, 2-26
Date flag, 3-3
DB2 catalog, 2-19
DB2 data bases, 1-5
DB2 logical records, 2-23
DB2 or SQL/DS tables, 2-18
DB2 table generation, 2-14
DBD name, 2-12
DD, 1-11
DD/DTF statement, 1-9, 1-11
debug level, 2-47
Decimal data, 2-21
DEDB, 2-23
definition glossaries, 1-7, 1-13, 2-5, 2-7
definition listing operations, 1-7
definition run, 1-3, 1-5
DEL, 2-8
DEL operation, 2-8
Device ID, 2-78
DF operator, 2-94, 2-101
direct dynamically synchronized database, 2-93
DL/I, 2-44
 calls, 2-49
 return code, 2-47
DL/I databases, 2-29, 2-92
DL/I path calls, 2-57
DMP, 2-8

documentation

Advantage VISION

Transact for CICS Environment Guide, 2-45,
2-50, 2-51, 2-69, 2-76, 3-7

Transact for CICS Installation Guide, 2-33,
2-48, 2-50, 2-63, 3-3

Transact Reference Guide, 2-10, 2-12, 2-29,
2-34, 2-38, 2-41, 2-78, 2-80, 2-81, 2-106, 2-112

Advantage VISION:Transact ASL Reference
Guide, 1-2, 1-8

COMLIB Installation Procedures, 2-118

DSDB I/O call tracing level, 2-5

DSDB loop tracing, 2-4

DUMP, 1-14

dynamic action group, 1-7

Dynamic partial field flags, 3-11

dynamic rule, 2-31

dynamically synchronized data base (DSDB), 2-4,
2-5

dynamically synchronized database
direct-read, 2-55

E

EC operator, 1-7, 2-86

EC operators, 1-13, 2-5

ED operation, 2-92

edit mask, 2-91

Edit/Validation operator is IF, 2-83

ELSE connector, 2-70

EN operator, 2-86

END, 2-38, 2-43

EOF, 1-14, 2-8

EQ operator, 2-40

ER statement, 1-6, 1-11, 2-9, 2-32, 2-39, 2-43, 2-68

error condition codes, 2-92

ERROR flag, 3-7

error handling, 2-102

establish request (ER), 2-9

ET operator, 1-7, 2-86

ET operators, 1-13, 2-5

execution tracing, 1-10, 2-1

execution tracing flags, 3-13

executor, 1-1

Extended Graphics (EGCS), 2-81

F

FALSE condition, 2-35

fast path, 1-5

fast path database, 2-30

FASTPATH, 2-60

FCT name, 2-12

FD delete entry, 1-5

FD statement, 1-10, 2-12, 2-18, 2-22, 2-25
Delete entry, 1-10

field errors, 2-92

File Control Table (FCT) name, 2-12

file definition statements, 1-5, 1-10, 2-12

Flag field, 2-67

flag field, 2-34

flags, 3-1

G

generator

functions, 1-1

use of COMLIB subsystem, 1-2

GETMAIN area, 2-59

glossary listing

type, 2-74

GOSC flag, 2-39, 3-8

H

highlighting, 2-101

HIGHVALU flag, 3-11

I

IF operator, 1-7, 2-93

IMS, 1-5

 TM TRANSACT macro, 2-76

IMS database, 2-12

IMS TM, 1-9, 2-44, 2-46, 2-50

 handles, 2-47

IMS TM environment, 1-1

INIT, 1-14

intermediate structure loop tracing, 2-4

INVFIELD flag, 3-15

ISAM file, 2-14

 defining, 2-12

ISDATE date format, 3-1

ISDATE flag, 3-3

IT statement, 2-15

IT statements, 1-4

item tracking, 2-15

item tracking statements, 1-4

ITEMTRAK flag, 2-16

J

Julian flag, 3-3

L

L0 statement, 1-5, 2-23, 2-25, 2-31, 2-53

LCHILD, 2-57

LFG, 2-7

LFN, 2-7

LFV, 2-7

LIBOPNA keyword, 1-8, 2-47

Library Open entry, 1-8

Library Open specification, 2-9

LIN, 2-7

LKEDCALL, 2-60

LKEDCALL programs, 2-48

LKEDCALL specification, 2-44

LKEDCIS keyword, 2-48

LKEDIMS keyword, 2-48

Ln statement, 2-17

LNUMBER flag, 2-41, 2-71

logic level, 2-19, 2-33, 2-63, 2-84

logical relationship statement, 2-18

LOWVALU flag, 3-11

LR statement, 2-18, 2-19, 2-20, 2-21, 2-31

LS statement, 2-22, 2-64

LSG, 2-7

LSN, 2-7

LST, 2-8

LSTART flag, 2-41, 2-71

M

M5 program, 1-3

M5LIB, 2-117

M5MAP, 1-11

M5MAP data set, 1-9

M5PARM, 1-8, 2-9, 2-16, 2-33, 2-47, 2-48, 2-63, 2-74, 3-3

 Device Table, 2-78

M5PUNCH, 2-8

M5PUNCH data set, 1-9

M5SSOUT, 2-48

M5SSOUT data set, 1-13, 2-7

M5TPTBL, 1-9

M5UTIL program, 1-3, 2-115

main storage database (MSDB), 2-26

MANUAL, 2-61

MAPSET, 2-77
MAPSET name, 2-73
MARKLIBP
 ITEMTRAK flag, 2-16
MARKV program, 1-3
MERG, 1-14
message operator, 2-35
message processing, 2-50
MFS source statements, 1-5, 1-11
MFS statements, 1-4
MNUMBER flag, 2-42, 2-71
MS operator, 2-34, 2-94
MSDB databases, 2-29
MSDB fast path database, 2-26
MSG command, 2-49
MSGLINE flag, 3-10
MSGLINE flag field, 2-78, 2-80
MSTART flag, 2-42, 2-71
MULTILIB, 2-61

N

named patterns, 3-15
NC operator, 2-88
NE operator, 2-40
NEXT FUNCTION, 2-39
NM operator, 2-88
NOROLLB literal, 2-34
NOROLLBACK option, 2-49
NS END conditional branch operator, 2-35
Null Segment Processing entry, 2-106

O

OC operator, 2-76
OC operators, 1-13, 2-5

OC statement, 2-11
OPERCL flag, 3-8
OPERID flag, 3-8
OR connector, 2-33
OR relationship, 2-19
OS operator, 1-6, 1-10, 2-11, 2-64, 2-67
OS operators, 1-8, 1-11, 1-12, 1-13, 2-5
OUTLIM parameter, 2-76
output edit/length, 2-30

P

PA key, 2-50
PAGENO flag, 3-11
partial field, 2-94
partial field specifications, 2-71
pattern validation, 2-90
PCB mask, 2-14, 2-25, 2-26, 2-30, 2-44, 2-53, 2-55, 2-56
 field, 2-34
PCB mask field, 2-64, 2-67
PCB masks, 1-5, 1-9, 2-12
PL/I, 2-43, 2-44
PNUMBER flag, 2-42, 2-71
PPT table entries, 1-9
PR statement, 2-11, 2-32, 2-33, 2-35, 2-38, 2-40, 2-45
 Call option, 2-43
procedure tracing, 2-3
PROCLIM, 2-51
Program Control Table (PCT), 2-77
PROGTYP flag, 3-8
PSB Name, 2-46
PSB statement, 1-9
pseudo file, 2-14, 2-25, 2-34, 2-38, 2-41, 2-53, 2-56, 2-64, 2-99
pseudo file definitions, 2-30

pseudo files, 1-5, 2-12
PSTART flag, 2-42, 2-71

R

RC statement, 1-4, 1-7, 1-8, 1-9, 1-11, 1-13, 2-9, 2-46, 2-52, 2-53, 2-54, 2-74
RECYCLE function, 2-70
recycle interaction, 2-114
RECYCLE parameter, 2-11
relational operator, 2-35
replacement operator, 2-35
request processing statements, 1-6, 1-11
REST, 1-14
RETURN-CODE keyword, 3-5
RETURNCD flag, 3-4
RF statement, 2-10, 2-24, 2-51, 2-53, 2-55
RF statements, 1-9
RFD, 1-14, 2-8
RNUMBER flag, 2-42, 2-71
root screen segment, 2-96
RP LKEDCALL statements, 2-48
RP MULTILIB statement, 1-13
RP MULTILIB statements, 1-9
RP statement, 2-60
RP statements, 1-4
RRG, 1-14, 2-8
RSD, 1-14, 2-8
RSTART flag, 2-42, 2-71
run control (RC) statement, 2-46
run control group, 1-4, 1-9, 1-13
Run Control statement, 1-7
run control statement, 1-4, 1-13
run name, 2-1
Run Name/Transaction Code entry, 2-77
run parameter statements, 1-4

S

S0 statement, 2-80, 2-97
SAV, 2-8
SAV operation, 2-8
SC statement, 1-11, 2-62
SC statements, 2-68
Scratch Pad Area, 2-58
screen control statements, 1-11
screen definition glossary, 2-34
screen definition statements, 1-7
Screen Layout (SL) statements, 2-74
screen layout statements, 1-7, 1-13, 2-5
screen synchronized data base (SSDB), 2-4
screen synchronized database, 2-53
screen synchronized database fields, 2-110
screen synchronized databases, 2-57
Screen/COMMAREA control statements, 1-10
Screen/COMMAREA Definition Statements, 1-11
screen/COMMAREA definition statements, 1-5
SCRNAMEI flag, 3-12
SCRNAMEO flag, 3-12
SD statement, 1-5, 1-7, 1-11, 2-46, 2-73, 2-82
SEGM statement, 2-22
segment action tracing, 2-3
segment definition, 2-22
segment errors, 2-92
segment operator (SG), 2-64
segment search arguments (SSAs), 2-31
SL statement, 1-7, 2-75
SL statements, 1-7, 1-13, 2-5, 2-74
Sn statement, 2-87
Sn statements, 2-82, 2-93
source statement listing, 2-48
source statement retrieval run, 2-8

source statement retrieval run (SSR), 1-13
SPA, 2-69
SQL databases, 2-92
SQL files, 2-12
SQL WHERE clause, 2-21
SQL/DS catalog, 2-19
SQL/DS logical records, 2-23
SQL/DS table generation, 2-14
SQL/DS tables, 2-18
SRCH parameter, 2-27
SS statement, 2-96
SSAs, 2-31
SSDB I/O call tracing level, 2-4
START parameter, 2-70
statement label, 2-38
statement type, 2-1
storage type, 2-58
Synchronizing Field 1, 2-56
Synchronizing Field 2, 2-55
SYNCOK flag, 3-8
SYNCONRETURN option, 3-8
SYNCPOINT, 3-8

T

TAPPFLOW, 2-3
target monitor, 2-52
TCONTROL, 2-2
TCONTROL flag, 3-14
TDLOOP, 2-4
TDSDBIO, 2-5
teleprocessing monitor, 1-4
temporary field, 2-34
Temporary Storage, 2-58
TERMID flag, 3-8

TF statement, 2-38, 2-112
TF statements, 1-12, 2-11
TFREQ specification, 1-12
TILOOP, 2-4
TIME flag, 3-4
TODAY flag, 2-16
Today flag, 3-3
TP monitor statements, 1-9
TPMON keyword, 2-52
TPROC, 2-3, 2-35
trace control, 2-2
Trace Control entry, 2-2
trace operator, 2-35
TRANCODE flag, 2-46, 2-69, 2-70, 2-76, 3-7
TRANSACT macro, 2-51
TRANSACT statement, 1-9
TRUE or FALSE condition, 2-35
TSEGACT, 2-3
TSSDBIO, 2-4
type 1 requests, 2-38, 2-53, 2-77
type 2 request, 2-10
type M request, 2-10, 2-34
type O request, 2-10, 2-39, 2-41, 2-67, 2-68, 3-12
type S request, 2-10, 2-39
type S requests, 1-8, 1-12

U

UC statement, 1-14, 2-115
UC statements, 1-14
updater ID, 2-16
USERID flag, 3-8
utility control, 2-115
utility control statement, 1-14
utility control statements, 1-14

V

validation patterns, 2-90, 3-15

VSAM databases, 2-30

VSAM file

 defining, 2-12

VSAM file (ESDS), 2-14

VSAM file (KSDS), 2-14

VSAM file (RRDS), 2-14

VSAM/ISAM files, 1-5

W

working storage, 1-5, 2-12, 2-30

Working storage area, 2-64

working storage area, 2-25, 2-26, 2-34

Working storage area field, 2-67

X

XATTR entry, 2-79

XDFLD specification, 2-27

XR operator, 1-6, 1-10, 2-11

XR operators, 1-8, 1-11, 1-12, 2-68

Z

z/OS, 2-52