

CICS® Transaction Server for VSE/ESA™



Application Programming Reference

Release 1

CICS® Transaction Server for VSE/ESA™



Application Programming Reference

Release 1

Note!

Before using this information and the product it supports, be sure to read the general information under “Notices” on page 441.

Fifth Edition (April 2006)

This edition applies to Release 1 of CICS Transaction Server for VSE/ESA, program number 5648-054, and to all subsequent versions, releases, and modifications until otherwise indicated in new editions. Make sure you are using the correct edition for the level of the product.

The CICS for VSE/ESA Version 2.3 edition remains applicable and current for users of CICS for VSE/ESA Version 2.3.

Order publications through your IBM representative or the IBM branch office serving your locality.

At the back of this publication is a page entitled “Sending your comments to IBM”. If you want to make any comments, please use one of the methods described there.

© Copyright International Business Machines Corporation 1982, 2006. All rights reserved.

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Preface	ix
What this book is about	ix
Who this book is for	ix
What you need to know to understand this book	ix
How to use this book	ix
Programming interface information	ix
Road map	ix
Notes on terminology	xi
Chapter 1. CICS API commands	1
Command format	1
CICS syntax notation used in this book	1
Argument values	2
VSE/ESA restrictions	4
CICS-value data areas (cvdas)	5
LENGTH options	5
NOHANDLE option	5
RESP and RESP2 options	5
DATASET	6
Possible ABEND AICA timeout	6
Commands by function	6
ABEND	10
ADDRESS	11
ADDRESS SET	13
ALLOCATE (APPC)	14
ALLOCATE (LUTYPE6.1)	16
ALLOCATE (MRO)	18
ASKTIME	19
ASSIGN	20
BIF DEEDIT	28
BUILD ATTACH (LUTYPE6.1)	29
BUILD ATTACH (MRO)	31
CANCEL	33
CHANGE PASSWORD	35
CHANGE TASK	36
CHKP (DL/I)	37
CONNECT PROCESS	38
CONVERSE (VTAM default)	40
CONVERSE (APPC)	40
CONVERSE (LUTYPE2/LUTYPE3)	41
CONVERSE (LUTYPE4)	41
CONVERSE (LUTYPE6.1)	42
CONVERSE (SCS)	42
CONVERSE (3270 logical)	43
CONVERSE (3600-3601)	43
CONVERSE (3600-3614)	44
CONVERSE (3650 interpreter)	45
CONVERSE (3650-3270)	45
CONVERSE (3650-3653)	46
CONVERSE (3650-3680)	46
CONVERSE (3767)	47
CONVERSE (3770)	47
CONVERSE (3790 full-function or inquiry)	48
CONVERSE (3790 3270-display)	48
CONVERSE (non-VTAM default)	52
CONVERSE (MRO)	52

CONVERSE (System/3)	53
CONVERSE (System/7)	53
CONVERSE (2260)	54
CONVERSE (2741)	55
CONVERSE (2770)	56
CONVERSE (2780)	56
CONVERSE (3270 display)	57
CONVERSE (3600 BTAM)	57
CONVERSE (3735)	59
CONVERSE (3740)	59
DELAY	63
DELETE	65
DELETEQ TD	69
DELETEQ TS	70
DEQ	71
DLET (DL/I)	73
DUMP	74
DUMP TRANSACTION	75
ENDBR	78
ENQ	80
ENTER TRACEID	82
ENTER TRACENUM	83
EXTRACT ATTACH (LUTYPE6.1)	85
EXTRACT ATTACH (MRO)	87
EXTRACT ATTRIBUTES (APPC)	89
EXTRACT ATTRIBUTES (MRO)	90
EXTRACT LOGONMSG	91
EXTRACT PROCESS	92
EXTRACT TCT	94
FORMATTIME	95
FREE	98
FREE (APPC)	99
FREE (LUTYPE6.1)	100
FREE (MRO)	101
FREEMAIN	102
GDS ALLOCATE	104
GDS ASSIGN	106
GDS CONNECT PROCESS	107
GDS EXTRACT ATTRIBUTES	109
GDS EXTRACT PROCESS	110
GDS FREE	111
GDS ISSUE ABEND	112
GDS ISSUE CONFIRMATION	113
GDS ISSUE ERROR	114
GDS ISSUE PREPARE	115
GDS ISSUE SIGNAL	116
GDS RECEIVE	117
GDS SEND	119
GDS WAIT	121
GETMAIN	122
GN (DL/I)	125
GNP (DL/I)	126
GU (DL/I)	127
HANDLE ABEND	128
HANDLE AID	130
HANDLE CONDITION	132
IGNORE CONDITION	134
ISRT (DL/I)	135
ISSUE ABEND	136
ISSUE ABORT	137

ISSUE ADD	139
ISSUE CONFIRMATION	141
ISSUE COPY (3270 display)	142
ISSUE COPY (3270 logical)	143
ISSUE DISCONNECT (default)	144
ISSUE DISCONNECT (LUTYPE6.1)	145
ISSUE END	146
ISSUE ENDFILE	148
ISSUE ENDOUTPUT	149
ISSUE EODS	150
ISSUE ERASE	151
ISSUE ERASEAUP	153
ISSUE ERROR	154
ISSUE LOAD	155
ISSUE NOTE	156
ISSUE PASS	157
ISSUE PREPARE	159
ISSUE PRINT	160
ISSUE QUERY	161
ISSUE RECEIVE	162
ISSUE REPLACE	164
ISSUE RESET	166
ISSUE SEND	167
ISSUE SIGNAL (APPC)	169
ISSUE SIGNAL (LUTYPE6.1)	170
ISSUE WAIT	171
JOURNAL	173
LINK	174
LOAD	178
LOAD (DL/I)	180
MONITOR	181
POINT	183
POP HANDLE	184
POST	185
PURGE MESSAGE	188
PUSH HANDLE	189
QUERY SECURITY	190
READ	192
READNEXT	197
READPREV	201
READQ TD	204
READQ TS	206
RECEIVE (VTAM default)	208
RECEIVE (APPC)	208
RECEIVE (LUTYPE2/LUTYPE3)	209
RECEIVE (LUTYPE4)	209
RECEIVE (LUTYPE6.1)	210
RECEIVE (3270 logical)	210
RECEIVE (3600 pipeline)	211
RECEIVE (3600-3601)	211
RECEIVE (3600-3614)	212
RECEIVE (3650)	212
RECEIVE (3767)	213
RECEIVE (3770)	214
RECEIVE (3790 full-function or inquiry)	214
RECEIVE (non-VTAM default)	217
RECEIVE (MRO)	217
RECEIVE (System/3)	218
RECEIVE (System/7)	218
RECEIVE (2260)	219

RECEIVE (2741)	219
RECEIVE (2980)	220
RECEIVE (3270 display)	222
RECEIVE (3600 BTAM)	222
RECEIVE (3735)	223
RECEIVE (3740)	223
RECEIVE (3790 3270-display)	224
RECEIVE MAP	228
RECEIVE PARTN	230
RELEASE	232
REPL (DL/I)	233
RESETBR	234
RETRIEVE	237
RETURN	240
REWRITE	242
ROUTE	244
SCHD (DL/I)	247
SEND (VTAM default)	248
SEND (APPC)	248
SEND (LUTYPE2/LUTYPE3)	249
SEND (LUTYPE4)	249
SEND (LUTYPE6.1)	250
SEND (SCS)	250
SEND (3270 logical)	251
SEND (3600 pipeline)	251
SEND (3600-3601)	252
SEND (3600-3614)	252
SEND (3650 interpreter)	253
SEND (3650-3270)	253
SEND (3650-3653)	254
SEND (3650-3680)	254
SEND (3767)	255
SEND (3770)	255
SEND (3790 full-function or inquiry)	256
SEND (3790 SCS)	256
SEND (3790 3270-display)	257
SEND (3790 3270-printer)	257
SEND (non-VTAM default)	261
SEND (MRO)	261
SEND (System/3)	262
SEND (System/7)	262
SEND (2260)	263
SEND (2741)	263
SEND (2980)	264
SEND (3270 display)	264
SEND (3600 BTAM)	265
SEND (3735)	265
SEND (3740)	266
SEND CONTROL	269
SEND MAP	273
SEND PAGE	278
SEND PARTNSET	281
SEND TEXT	282
SEND TEXT MAPPED	286
SEND TEXT NOEDIT	288
SIGNOFF	291
SIGNON	292
SPOOLCLOSE	294
SPOOLCLOSE REPORT	295
SPOOLOPEN ESCAPE	298

SPOOLOPEN INPUT	301
SPOOLOPEN MAPNAME	303
SPOOLOPEN OUTPUT	307
SPOOLOPEN REPORT (ASA, MCC or NOCC)	310
SPOOLOPEN REPORT (JCL)	315
SPOOLOPEN REPORT (SCS/T3270)	317
SPOOLOPEN RESUME	321
SPOOLREAD	323
SPOOLWRITE	324
SPOOLWRITE MAPNAME	326
SPOOLWRITE REPORT	329
START	331
START ATTACH	336
STARTBR	337
SUSPEND	340
SYNCPOINT	341
SYNCPOINT ROLLBACK	342
TERM (DL/I)	343
TRACE	344
UNLOCK	345
VERIFY PASSWORD	347
WAIT CONVID (APPC)	349
WAIT EVENT	350
WAIT EXTERNAL	352
WAIT JOURNAL	354
WAIT JOURNALNUM	355
WAIT SIGNAL	356
WAIT TERMINAL	357
WAITCICS	358
WRITE	360
WRITE JOURNALNUM	363
WRITE OPERATOR	365
WRITEQ TD	367
WRITEQ TS	369
XCTL	371
Appendix A. EXEC interface block	373
EIB fields	373
Appendix B. Codes returned by ASSIGN	381
ASSIGN TERMCODE	381
ASSIGN FCI	381
Appendix C. Translated code for CICS commands	383
COBOL	383
C	383
PL/I	383
Assembler language	384
Appendix D. Terminal control	387
Commands and options for terminals and logical units	387
BTAM programmable terminals	388
Teletypewriter programming	389
Display device operations	390
Appendix E. SAA Resource Recovery	393
SRRCMT	393
SRRBACK	393

Appendix F. Common Programming Interface Communications (CPI Communications)	395
Appendix G. API restrictions for distributed program link	397
Summary of the restricted API commands	397
List of API commands	397
Appendix H. CVDA numeric values	401
Appendix I. National language codes	403
Appendix J. BMS-related constants	405
Magnetic slot reader (MSR) control value constants, DFHMSRCA	407
MSR control byte values	407
Attention identifier constants, DFHAID	409
Appendix K. BMS macro summary	411
Mapset, map, and field definition	411
Partition set definition	411
DFHMDF	414
DFHMDI	421
DFHMSD	427
DFHPDI	433
DFHPSD	434
Bibliography	435
Books from VSE/ESA 2.4 base program libraries	436
Books from VSE/ESA 2.4 optional program libraries	438
Notices	441
Trademarks and service marks	442
Index	443

Preface

What this book is about

This book describes the CICS Transaction Server for VSE/ESA application programming interface. It contains **reference** information needed to prepare COBOL, C, PL/I, and assembler-language application programs, using CICS commands, to be executed under CICS. **Guidance** information is in the *CICS Application Programming Guide*. For information about debugging CICS applications, see the *CICS Problem Determination Guide*.

Who this book is for

The book is intended primarily for use by application programmers, but will also be useful to system programmers and systems analysts.

What you need to know to understand this book

We assume that you have some experience in writing programs in COBOL, C, PL/I, or S370 assembler language. The *CICS Application Programming Guide* will help you to design and write CICS applications using the commands described in this book.

How to use this book

This book is for reference. Each of the commands has a standard format, as follows:

- The syntax of the command
- A description of what the command does
- An alphabetical list of the options and their functions
- An alphabetical list of conditions, and their causes, that can occur during execution of a command.

Programming interface information

This book documents General-use Programming Interface and Associated Guidance Information and Product-sensitive Programming Interface and Associated Guidance Information provided by CICS.

General-use programming interfaces allow the customer to write programs that obtain the services of CICS.

Product-sensitive programming interfaces allow the customer installation to perform tasks such as diagnosing, modifying, monitoring, repairing, tailoring, or tuning of CICS. Use of such interfaces creates dependencies on the detailed design or implementation of the IBM software product. Product-sensitive programming interfaces should be used only for these specialized purposes. Because of their dependencies on detailed design and implementation, it is to be expected that programs written to such interfaces may need to be changed in order to run with new product releases or versions, or as a result of service.

Road map

If you want to...	Refer to...
Know about CICS API command format	Chapter 1, "CICS API commands" on page 1.
Look up EXEC interface block (EIB) fields	Appendix A, "EXEC interface block" on page 373.
Look up codes returned by ASSIGN	Appendix B, "Codes returned by ASSIGN" on page 381.

Table 1 (Page 2 of 2). Getting started road map

If you want to...	Refer to...
Read about translated code for CICS API commands	Appendix C, "Translated code for CICS commands" on page 383.
Read about CICS terminal control	Appendix D, "Terminal control" on page 387.
Read about SAA Resource Recovery	Appendix E, "SAA Resource Recovery" on page 393.
Read about the Common Programming Interface for Communications (CPIC)	Appendix F, "Common Programming Interface Communications (CPI Communications)" on page 395.
Know about API restrictions for Distributed Program Link (DPL)	Appendix G, "API restrictions for distributed program link" on page 397.
Look up CVDA numeric values	Appendix H, "CVDA numeric values" on page 401.
Look up national language codes	Appendix I, "National language codes" on page 403.
Look up BMS related constants	Appendix J, "BMS-related constants" on page 405.
Define BMS mapsets, maps, and fields	Appendix K, "BMS macro summary" on page 411.

Notes on terminology

The terms listed in Table 2 are commonly used in the CICS Transaction Server for VSE/ESA Release 1 library. See the *CICS Glossary* for a comprehensive definition of terminology.

<i>Table 2 (Page 1 of 2). Commonly used words and abbreviations in CICS Transaction Server for VSE/ESA Release 1</i>	
Term	Definition (and abbreviation if appropriate)
\$(the dollar symbol)) In the character sets and programming examples given in this book, the dollar symbol (\$) is used as a national currency symbol and is assumed to be assigned the EBCDIC code point X'5B'. In some countries a different currency symbol, for example the pound symbol (£), or the yen symbol (¥), is assigned the same EBCDIC code point. In these countries, the appropriate currency symbol should be used instead of the dollar symbol.
BSM	BSM is used to indicate the basic security management supplied as part of the VSE/ESA product. It is RACROUTE-compliant, and provides the following functions: <ul style="list-style-type: none"> • Signon security • Transaction attach security
C	The C programming language
CICSplex	A CICSplex consists of two or more regions that are linked using CICS intercommunication facilities. Typically, a CICSplex has at least one terminal-owning region (TOR), more than one application-owning region (AOR), and may have one or more regions that own the resources accessed by the AORs
CICS Data Management Facility	The new CICS Transaction Server for VSE/ESA Release 1 facility to which all statistics and monitoring data is written, generally referred to as "DMF"
CICS/VSE	The CICS product running under the VSE/ESA operating system, frequently referred to as simply "CICS"
COBOL	The COBOL programming language
DB2 for VSE/ESA	Database 2 for VSE/ESA which was previously known as "SQL/DS".
ESM	ESM is used to indicate a RACROUTE-compliant external security manager that supports some or all of the following functions: <ul style="list-style-type: none"> • Signon security • Transaction attach security • Resource security • Command security • Non-terminal security • Surrogate user security • MRO/ISC security (MRO, LU6.1 or LU6.2) • FEPI security.
FOR (file-owning region)—also known as a DOR (data-owning region)	A CICS region whose primary purpose is to manage VSAM and DAM files, and VSAM data tables, through function provided by the CICS file control program.

Table 2 (Page 2 of 2). Commonly used words and abbreviations in CICS Transaction Server for VSE/ESA Release 1

Term	Definition (and abbreviation if appropriate)
IBM C for VSE/ESA	The Language Environment version of the C programming language compiler. Generally referred to as "C/VSE".
IBM COBOL for VSE/ESA	The Language Environment version of the COBOL programming language compiler. Generally referred to as "COBOL/VSE".
IBM PL/I for VSE/ESA	The Language Environment version of the PL/I programming language compiler. Generally referred to as "PL/I VSE".
IBM Language Environment for VSE/ESA	The common runtime interface for all LE-conforming languages. Generally referred to as "LE/VSE".
PL/I	The PL/I programming language
VSE/POWER	Priority Output Writers Execution processors and input Readers. The VSE/ESA spooling subsystem which is exploited by the report controller.
VSE/ESA System Authorization Facility	The new VSE facility which enables the new security mechanisms in CICS TS for VSE/ESA R1, generally referred to as "SAF"
VSE/ESA Central Functions component	The new name for the VSE Advanced Function (AF) component
VSE/VTAM	"VTAM"

Chapter 1. CICS API commands

This book shows the syntax of each command, describes the purpose and format of each command and its options, and gives a list of the conditions that can arise during the execution of a command.

Note: The INQUIRE and SET commands of the application programming interface (API) are primarily for the use of the system programmer; they are not described in this book. For details of the commands, refer to the *CICS System Programming Reference*.

For information about translating the commands, see the *CICS Application Programming Guide* for translator options, and the *CICS System Definition Guide* for the JCL.

Command format

The general format of a CICS command is EXECUTE CICS (or EXEC CICS) followed by the name of the required **command**, and possibly by one or more **options**, as follows:

```
EXEC CICS command option(arg)....
```

where:

- command** describes the operation required (for example, READ).
- option** describes any of the many optional facilities available with each function. Some options are followed by an argument in parentheses. You can write options (including those that require arguments) in any order.
- arg** (**short for argument**) is a value such as “data-value” or “name”. A “data-value” can be a constant, this means that an argument that sends data to CICS is generally a “data-value”. An argument that receives data from CICS must be a “data-area”.
Some arguments described as “data-area” can both send and receive data. In these cases, you must ensure that the “data-area” is not in protected storage.

An example of a CICS command is as follows:

```
EXEC CICS READ
      FILE('FILEA')
      INTO(FILEA)
      RIDFLD(KEYNUM)
      UPDATE
```

You must add the appropriate end-of-command delimiter; see “CICS syntax notation used in this book.” Similarly, DL/I services have the format:

```
EXEC DLI function option(arg) ....
```

CICS syntax notation used in this book

In the CICS books, CICS commands are presented in a standard way.

The “EXEC CICS” that always precedes each command’s keyword is not included; nor is the “END-EXEC” statement used in COBOL or the semicolon (;) used in PL/I and C that you must code at the end of each CICS command. In the C language, a null character can be used as an end-of-string marker, but CICS does not recognize this; you must therefore never have a comma or period followed by a space (X'40') in the middle of a coding line.

You interpret the syntax by following the arrows from left to right. The conventions are:

Symbol	Action
	A set of alternatives—one of which you must code.
	A set of alternatives—one of which you may code.
	A set of alternatives—any number of which you may code once.
	Alternatives where A is the default.
	Use with the named section in place of its name.
Punctuation and uppercase characters	Code exactly as shown.
Lowercase characters	Code your own text, as appropriate (for example, name).

For example, with READ FILE(filename) you must code READ FILE and () unchanged, but are free to code any valid text string to mean the name of the file.

Argument values

The parenthesized argument values that follow options in a CICS command are specified as follows:

- data-value
- data-area
- cvda (CICS-value data area)
- ptr-value
- ptr-ref
- name
- label
- hhmss
- filename
- systemname

COBOL argument values

The argument values can be replaced as follows:

- “data-value” can be replaced by any COBOL data name of the correct data type for the argument, or by a constant that can be converted to the correct type for the argument. The data type can be specified as one of the following:
 - Halfword binary — PIC S9(4) COMP
 - Fullword binary — PIC S9(8) COMP
 - Character string — PIC X(n) where “n” is the number of bytes
- “data-area” can be replaced by any COBOL data name of the correct data type for the argument. The data type can be specified as one of the following:
 - Halfword binary — PIC S9(4) COMP
 - Fullword binary — PIC S9(8) COMP
 - Character string — PIC X(n) where “n” is the number of bytes

Where the data type is unspecified, “data-area” can refer to an elementary or group item.

- “cvda” is described in “CICS-value data areas (cvdas)” on page 5.
- “ptr-value” can be replaced by a COBOL pointer variable.
- “ptr-ref” can be replaced by a COBOL pointer variable, or an ADDRESS special register.
- “name” can be replaced by either of the following:
 - A character string in single quotation marks (that is, a nonnumeric literal). If this is shorter than the required length, it is padded with blanks.
 - A COBOL data area with a length equal to the length required for the name. The value in “data-area” is the name to be used by the argument. If “data-area” is shorter than the required length, the excess characters are undefined.

“filename”, as used in FILE(filename), specifies the name of the file. It has 1–7 characters from A–Z, 0–9, \$, @, and #. (Lowercase characters are converted to

uppercase.) If replaced by a COBOL data area, its length must be eight characters, and names shorter than eight characters must be padded with blanks.

“systemname”, as used in SYSID(systemname), specifies the name of the system the request is directed to. It has 1–4 characters from A–Z, 0–9, \$, @, and #. (Lowercase characters are converted to uppercase.)

- “label” can be replaced by any COBOL paragraph name or a section name.
- “hhmss” can be replaced by a decimal constant or by a data name of the form PIC S9(7) COMP-3. The value must be of the form 0HHMMSS+ where:

HH represents hours from 00 through 99.

MM represents minutes from 00 through 59.

SS represents seconds from 00 through 59.

There is no need to code the LENGTH option unless you want the program to read or write data of a length different from that of the referenced variable.

C argument values

The argument values can be replaced as follows:

- “data-value” can be replaced by any C expression that can be converted to the correct data type for the argument. The data type can be specified as one of the following:
 - Halfword binary — short int
 - Fullword binary — long int
 - Character string — char[n] where “n” is the number of bytes
- “data-value” includes “data-area” as a subset.
- “data-area” can be replaced by any C data reference that has the correct data type for the argument. The data type can be specified as one of the following:
 - Halfword binary — short int
 - Fullword binary — long int
 - Character string — char[n] where “n” is the number of bytes

If the data type is unspecified, “data-area” can refer to a scalar data type, array, or structure. The reference must be to contiguous storage.

- “cvda” is described in “CICS-value data areas (cvdas)” on page 5.
- “ptr-value” (which includes “ptr-ref” as a subset) can be replaced by any C expression that can be converted to an address.
- “ptr-ref” can be replaced by any C pointer type reference.
- “name” can be replaced by either of the following:
 - A character string in double quotation marks (that is, a literal constant).

- A C expression or reference whose value can be converted to a character array with a length equal to the maximum length allowed for the name. The value of the character array is the name to be used by the argument.

“filename”, as used in FILE(filename), specifies the name of the file. It has 1–7 characters from A–Z, 0–9, \$, @, and #. (Lowercase characters are converted to uppercase.) If replaced by a C expression or reference, its length must be eight characters, and names shorter than eight/characters must be padded with blanks.

“systemname”, as used in SYSID(systemname), specifies the name of the system the request is directed to. It has 1–4 characters from A–Z, 0–9, \$, @, and #. (Lowercase characters are converted to uppercase.)

- “label” is not supported in the C language.
- “hhmss” is not supported in the C language.

Many commands involve the transfer of data between the application program and CICS.

In most cases, the LENGTH option must be specified if SET is used; the syntax of each command and its associated options show whether or not this rule applies.

PL/I argument values

The argument values can be replaced as follows:

- “data-value” can be replaced by any PL/I expression that can be converted to the correct data type for the argument. The data type can be specified as one of the following:
 - Halfword binary — FIXED BIN(15)
 - Fullword binary — FIXED BIN(31)
 - Character string — CHAR(n) where “n” is the number of bytes
- “data-value” includes “data-area” as a subset.
- “data-area” can be replaced by any PL/I data reference that has the correct data type for the argument. The data type can be specified as one of the following:
 - Halfword binary — FIXED BIN(15)
 - Fullword binary — FIXED BIN(31)
 - Character string — CHAR(n) where “n” is the number of bytes

If the data type is unspecified, “data-area” can refer to an element, array, or structure; for example, FROM(P->STRUCTURE) LENGTH(LNG). The reference must be to connected storage.

The data area must also have the correct PL/I alignment attribute: ALIGNED for binary items, and UNALIGNED for strings.

If you use a varying data string without an explicit length, the data passed begins with two length bytes, and its length is the maximum length declared for the string. If you explicitly specify a length in the command, the data

passed has this length; that is, the two length bytes followed by data up to the length you specified.

- “cvda” is described in “CICS-value data areas (cvdas)” on page 5.
- “ptr-value” (which includes “ptr-ref” as a subset) can be replaced by any PL/I expression that can be converted to POINTER.
- “ptr-ref” can be replaced by any PL/I reference of type POINTER ALIGNED.
- “name” can be replaced by either of the following:
 - A character string in single quotation marks (that is, a literal constant).
 - A PL/I expression or reference whose value can be converted to a character string with a length equal to the maximum length allowed for the name. The value of the character string is the name to be used by the argument.

“filename”, as used in FILE(filename), specifies the name of the file. It has 1–7 characters from A–Z, 0–9, \$, @, and #. (Lowercase characters are converted to uppercase.) If replaced by a PL/I expression or reference, its length must be eight characters, and names shorter than eight characters must be padded with blanks.

“systemname”, as used in SYSID(systemname), specifies the name of the system the request is directed to. It has characters from A–Z, 0–9, \$, @, and #. (Lowercase characters are converted to uppercase.)

- “label” can be replaced by any PL/I expression whose value is a label.
- “hhmss” can be replaced by a decimal constant or an expression that can be converted to a FIXED DECIMAL(7,0). The value must be of the form OHHMMSS+ where:

HH	represents hours from 00 through 99.
MM	represents minutes from 00 through 59.
SS	represents seconds from 00 through 59.

If the UNALIGNED attribute is added to the ENTRY declarations generated by the CICS translator by a DEFAULT DESCRIPTORS statement, data-area or pointer-reference arguments to CICS commands must also be UNALIGNED. Similarly for the ALIGNED attribute, data-area or pointer-reference arguments must be ALIGNED.

Many commands involve the transfer of data between the application program and CICS.

In most cases, the length of the data to be transferred must be provided by the application program. However, if a data area is specified as the source or target, it is not necessary to provide the length explicitly, because the command-language translator generates a default length value of either STG(data-area) or CSTG(data-area), as appropriate.

Assembler-language argument values

In general, an argument may be either the address of the data or the data itself (in assembler-language terms, either a relocatable expression or an absolute expression).

A relocatable expression must not contain unmatched brackets (outside quotation marks) or unmatched quotation marks (apart from length-attribute references). If this rule is obeyed, any expression can be used, including literal constants, such as =AL2(100), forms such as 20(0,R11), and forms that use the macro-replacement facilities.

An absolute expression must be a single term that is either a length-attribute reference, or a self-defining constant.

Care must be taken with equated symbols, which should be used only when referring to registers (pointer references). If an equated symbol is used for a length, for example, it is treated as the address of the length and an unpredictable error occurs.

The argument values can be replaced as follows:

- “data-value” can be replaced by a relocatable expression that is an assembler-language reference to data of the correct type for the argument, or by a constant of the correct type for the argument.
- “data-area” can be replaced by a relocatable expression that is an assembler-language reference to data of the correct type for the argument.
- “cvda” is described in “CICS-value data areas (cvdas)” on page 5.
- “ptr-value” can be replaced by an absolute expression that is an assembler-language reference to a register.
- “ptr-ref” can be replaced by an absolute expression that is an assembler-language reference to a register.
- “name” can be replaced **either** by a character string in single quotation marks, **or** by an assembler-language relocatable expression reference to a character string. The length is equal to the maximum length allowed for the name. The value of the character string is the name to be used by the argument.

“filename”, as used in FILE(filename), specifies the name of the file. It has 1–7 characters from A–Z, 0–9, \$, @, and #. (Lowercase characters are converted to uppercase.) If replaced by an assembler-language relocatable expression, its length must be eight characters, and names shorter than eight characters must be padded with blanks.

“systemname”, as used in SYSID(systemname), specifies the name of the system the request is directed to. It has 1–4 characters from A–Z, 0–9, \$, @, and #. (Lowercase characters are converted to uppercase.)

- “label” refers to a destination address to which control is transferred. It can be replaced by the label of the destination instruction or by the label of an address

constant for the destination. This constant must not specify a length.

You can also use the expression =A(dest) where “dest” is a relocatable expression denoting the destination.

For example, the following commands are equivalent:

```
HANDLE CONDITION ERROR(DEST)
HANDLE CONDITION ERROR(ADCON)
HANDLE CONDITION ERROR(=A(DEST))
:
DEST BR 14
ADCON DC A(DEST)
```

- “hhmmss” can be replaced by a self-defining decimal constant, or an assembler-language reference to a field defined as PL4. The value must be of the form OHHMMSS+ where:

HH represents hours from 00 through 99
MM represents minutes from 00 through 59
SS represents seconds from 00 through 59.

Many commands involve the transfer of data between the application program and CICS.

In most cases, the length of the data to be transferred must be provided by the application program. However, if a data area is specified as the source or target, it is not necessary to provide the length explicitly, because the command-language translator generates a default length.

For example:

```
xxx DC CL8
.
.
EXEC CICS ... LENGTH(L'xxx)
```

VSE/ESA restrictions

The following general restrictions apply to all CICS commands:

The restrictions that apply to CICS commands that access user data:

- The program must be in primary addressing mode when invoking any CICS service. The primary address space must be the home address space. All parameters passed to CICS must reside in the primary address space.
- CICS does not always preserve access registers across CICS commands or macro invocations. If your program uses access registers, it should save them before invoking a CICS service, and restore them before reusing them.

CICS-value data areas (cvdas)

There are options on a number of commands that describe or define a resource. CICS supplies, in CICS-value data areas, the values associated with these options. The options are shown in the syntax of the commands with the term “cvda” in parentheses.

You pass a cvda value in two different ways:

- You can assign a cvda value with the translator routine DFHVALUE. This allows you to change a cvda value in the program as the result of other run-time factors.

For example:

```
MOVE DFHVALUE(NOTPURGEABLE) TO AREA-A.  
EXEC CICS WAIT EXTERNAL ECBLIST() NUMEVENTS()  
                PURGEABILITY(AREA-A)
```

- If the required action is always the same, you can declare the value directly.

For example:

```
EXEC CICS WAITCICS ECBLIST() NUMEVENTS() PURGEABLE
```

You receive a cvda value by defining a fullword binary data area and then testing the returned value with the translator routine DFHVALUE. For example:

```
EXEC CICS CONNECT PROCESS .... STATE(AREA-A)  
IF AREA-A = DFHVALUE(ALLOCATED) ....  
IF AREA-A = DFHVALUE(CONFFREE) ....
```

LENGTH options

In COBOL, PL/I, and assembler language, the translator defaults certain lengths, if the NOLENGTH translator option is not specified. This means they are optional in programs that specify data areas.

In C, all LENGTH options must be specified.

When a CICS command offers the LENGTH option, it is generally expressed as a signed halfword binary value. This puts a theoretical upper limit of 32 763 bytes on LENGTH. In practice (depending on issues of recoverability, function shipping, and so on), the achievable upper limit varies from command to command, but is somewhat less than this theoretical maximum.

To be safe, do not let the value assigned to the length option for any CICS command exceed 24KB. For journal commands, the restrictions apply to the sum of LENGTH and PFXLENG values.

For *temporary storage*, *transient data*, and *file control* commands, the data set definitions may themselves impose further restrictions.

NOHANDLE option

You can use the NOHANDLE option with any command to specify that you want no action to be taken for any condition or AID resulting from the execution of that command. For further information about the NOHANDLE option, see the *CICS Application Programming Guide*.

Note that using the C language implies NOHANDLE on all commands.

RESP and RESP2 options

You can use the RESP option with any command to test whether a condition was raised during its execution. With some commands, when a condition can be raised for more than one reason, you can, if you have already specified RESP, use the RESP2 option to determine exactly why a condition occurred.

RESP(xxx)

“xxx” is a user-defined fullword binary data area. On return from the command, it contains a value corresponding to the condition that may have been raised, or to a normal return, that is, xxx=DFHRESP(NORMAL). You can test this value by means of DFHRESP, as follows:

```
EXEC CICS WRITEQ TS FROM(abc)  
                QUEUE(qname)  
                RESP(xxx)  
                RESP2(yyy)  
.  
.  
IF xxx=DFHRESP(NOSPACE) THEN ...
```

The above form of DFHRESP applies to both COBOL and PL/I.

An example of a similar test in C:

```
switch (xxx) {  
  case DFHRESP(NORMAL) : break;  
  case DFHRESP(INVREQ) : Invreq_Cond();  
                        break;  
  default               : Errors();  
}
```

An example of a similar test in assembler language:

```
CLC   xxx,DFHRESP(NOSPACE)
```

which the translator changes to:

```
CLC   xxx,=F'18'
```

As the use of RESP implies NOHANDLE, you must be careful when using RESP with the RECEIVE command, because NOHANDLE overrides the HANDLE AID command as well as the HANDLE CONDITION command, with the result that PF key responses are ignored.

RESP2(yyy)

“yyy” is a user-defined fullword binary data area. On return from the command, it contains a value that further qualifies the response to certain commands. Unlike the RESP values, RESP2 values have no associated symbolic names and there is no translator built-in function corresponding to DFHRESP, so you must test the fullword binary value itself.

RESP2 values are given in the description of each command that returns them.

DATASET

FILE has now replaced the DATASET keyword, although DATASET is still accepted to support existing applications.

Possible ABEND AICA timeout

A task can be abended by CICS if it has been running for longer than the runaway time, ICVR, specified in the system initialization table (SIT). This abend can be prevented by coding an EXEC CICS SUSPEND command in the application. This causes the dispatcher to suspend the task that issued the request and allow any task of higher priority to run. If there is no task ready to run, the program that issued the suspend is resumed. For further information about abend AICA, see the *CICS Problem Determination Guide*.

Commands by function

The following is a list of EXEC CICS commands categorized according to the function they perform.

Abend support

ABEND
HANDLE ABEND

APPC basic conversation

GDS ALLOCATE
GDS ASSIGN
GDS CONNECT PROCESS
GDS EXTRACT ATTRIBUTES
GDS EXTRACT PROCESS
GDS FREE
GDS ISSUE ABEND
GDS ISSUE CONFIRMATION
GDS ISSUE ERROR
GDS ISSUE PREPARE
GDS ISSUE SIGNAL
GDS RECEIVE
GDS SEND
GDS WAIT

APPC mapped conversation

ALLOCATE (APPC)
CONNECT PROCESS
CONVERSE (APPC)
EXTRACT ATTRIBUTES (APPC)
EXTRACT PROCESS
FREE (APPC)
ISSUE ABEND
ISSUE CONFIRMATION
ISSUE ERROR
ISSUE PREPARE
ISSUE SIGNAL (APPC)
RECEIVE (APPC)
SEND (APPC)
WAIT CONVID

Authentication

CHANGE PASSWORD
SIGNOFF
SIGNON
VERIFY PASSWORD

Batch data interchange

ISSUE ABORT
ISSUE ADD
ISSUE END
ISSUE ERASE
ISSUE NOTE
ISSUE QUERY
ISSUE RECEIVE
ISSUE REPLACE
ISSUE SEND
ISSUE WAIT

BMS

PURGE MESSAGE
RECEIVE MAP
RECEIVE PARTN
ROUTE
SEND CONTROL
SEND MAP
SEND PAGE
SEND PARTNSET
SEND TEXT
SEND TEXT MAPPED
SEND TEXT NOEDIT

Built-in functions

BIF DEEDIT

Console support

WRITE OPERATOR

Diagnostic services

DUMP*
DUMP TRANSACTION
ENTER TRACEID*
ENTER TRACENUM
TRACE*

DL/I services

CHKP (CHECKPOINT)
DLET (DELETE)
GN (GET NEXT)
GNP (GET NEXT IN PARENT)
GU (GET UNIQUE)
ISRT (INSERT)
LOAD
REPL (REPLACE)
SCHD (SCHEDULE)
TERM (TERMINATE)

Environment services

ADDRESS
ADDRESS SET
ASSIGN

Exception support

HANDLE CONDITION
IGNORE CONDITION
POP HANDLE
PUSH HANDLE

File control

DELETE
ENDBR
READ
READNEXT
READPREV
RESETBR
REWRITE
STARTBR
UNLOCK
WRITE

Interval control

ASKTIME
CANCEL
DELAY
FORMATTIME
POST
RETRIEVE
START
WAIT EVENT

Journaling

JOURNAL*
WAIT JOURNAL*
WAIT JOURNALNUM
WRITE JOURNALNUM

Monitoring

MONITOR

Program control

LINK
LOAD
RELEASE
RETURN
XCTL

* All commands marked by ** have been superseded by new commands that should be used in all new applications. They are still supported for migration purposes.

Report controller

SPOOLCLOSE REPORT
SPOOLOPEN ESCAPE
SPOOLOPEN MAPNAME
SPOOLOPEN REPORT
SPOOLOPEN REPORT (JCL)
SPOOLOPEN REPORT (SCS)
SPOOLOPEN REPORT (ASA, MCC, NOCC)
SPOOLOPEN RESUME
SPOOLWRITE MAPNAME
SPOOLWRITE REPORT

Security

QUERY SECURITY

Spool Interface (POWER)

SPOOLCLOSE
SPOOLOPEN INPUT
SPOOLOPEN OUTPUT
SPOOLREAD
SPOOLWRITE

Storage control

FREEMAIN
GETMAIN

Syncpoint

SYNCPOINT
SYNCPOINT ROLLBACK

Task control

CHANGE TASK
DEQ
ENQ
START ATTACH
SUSPEND
WAIT EXTERNAL
WAITCICS

Temporary storage control

DELETEQ TS
READQ TS
WRITEQ TS

Terminal control

ALLOCATE (LUTYPE6.1)
ALLOCATE (MRO)
BUILD ATTACH (LUTYPE6.1)
BUILD ATTACH (MRO)
CONVERSE (default)
CONVERSE (LUTYPE2/LUTYPE3)
CONVERSE (LUTYPE4)
CONVERSE (LUTYPE6.1)
CONVERSE (MRO)
CONVERSE (SCS)
CONVERSE (System/3)
CONVERSE (System/7)
CONVERSE (2260)
CONVERSE (2741)
CONVERSE (3270 display)
CONVERSE (3270 logical)
CONVERSE (3600 BTAM)
CONVERSE (3600-3601)
CONVERSE (3600-3614)
CONVERSE (3650 interpreter)
CONVERSE (3650-3270)
CONVERSE (3650-3653)
CONVERSE (3650-3680)
CONVERSE (3735)
CONVERSE (3740)
CONVERSE (3767)
CONVERSE (3770)
CONVERSE (3790 full-function or inquiry)
CONVERSE (3790 3270-display)
EXTRACT ATTACH (LUTYPE6.1)
EXTRACT ATTACH (MRO)
EXTRACT ATTRIBUTES (MRO)
EXTRACT LOGONMSG
EXTRACT TCT
FREE (LUTYPE6.1)
FREE
FREE (MRO)
HANDLE AID
ISSUE COPY (3270 display)
ISSUE COPY (3270 logical)
ISSUE DISCONNECT (default)
ISSUE DISCONNECT (LUTYPE6.1)
ISSUE ENDFILE
ISSUE ENDOUTPUT
ISSUE EODS
ISSUE ERASEAUP
ISSUE LOAD
ISSUE PASS
ISSUE PRINT
ISSUE RESET
ISSUE SIGNAL (LUTYPE6.1)
POINT
RECEIVE (default)
RECEIVE (LUTYPE2/LUTYPE3)
RECEIVE (LUTYPE4)
RECEIVE (LUTYPE6.1)
RECEIVE (MRO)
RECEIVE (System/3)

RECEIVE (System/7)
RECEIVE (2260)
RECEIVE (2741)
RECEIVE (2980)
RECEIVE (3270 display)
RECEIVE (3270 logical)
RECEIVE (3600 BTAM)
RECEIVE (3600 pipeline)
RECEIVE (3600-3601)
RECEIVE (3600-3614)
RECEIVE (3650)
RECEIVE (3735)
RECEIVE (3740)
RECEIVE (3767)
RECEIVE (3770)
RECEIVE (3790 full-function or inquiry)
RECEIVE (3790 3270-display)
SEND (default)
SEND (LUTYPE2/LUTYPE3)
SEND (LUTYPE4)
SEND (LUTYPE6.1)
SEND (MRO)
SEND (SCS)
SEND (System/3)
SEND (System/7)
SEND (2260)
SEND (2741)
SEND (2980)
SEND (3270 display)
SEND (3270 logical)
SEND (3600 BTAM)
SEND (3600 pipeline)
SEND (3600-3601)
SEND (3600-3614)
SEND (3650 interpreter)
SEND (3650-3270)
SEND (3650-3653)
SEND (3650-3680)
SEND (3735)
SEND (3740)
SEND (3767)
SEND (3770)
SEND (3790 full-function or inquiry)
SEND (3790 SCS)
SEND (3790 3270-display)
SEND (3790 3270-printer)
WAIT SIGNAL
WAIT TERMINAL

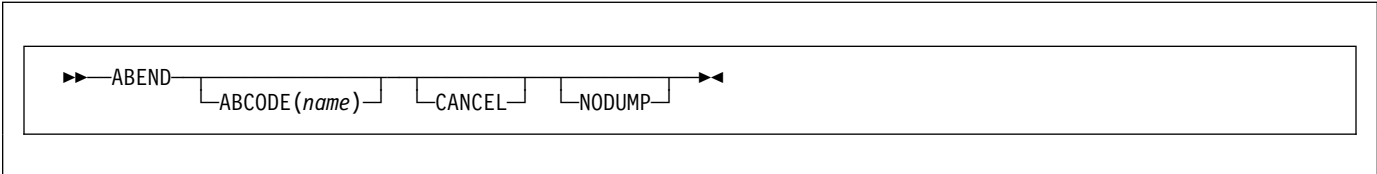
Transient data

DELETEQ TD
READQ TD
WRITEQ TD

ABEND

ABEND

Terminate a task abnormally.



Description

ABEND terminates a task abnormally.

The main storage associated with the terminated task is released; optionally, a transaction dump of this storage can be obtained.

Options

ABCODE(*name*)

specifies that main storage related to the task that is being terminated is to be dumped. The ABCODE is used as a transaction dumpcode to identify the dump. The name should have four characters and should not contain any leading or imbedded blanks. If ABCODE is not coded, the dump is identified by ????.

Do not start the name with the letter A, because this is reserved for CICS itself.

CANCEL

specifies that exits established by HANDLE ABEND commands are to be ignored. An ABEND CANCEL command cancels all exits at any level in the task (and terminates the task abnormally). If the TRAP(ON) LE/VSE run-time option has been specified, an abnormal termination exit is established by LE/VSE. This exit is revoked by the CANCEL option.

NODUMP

allows you to request an abend without causing a dump to be taken.

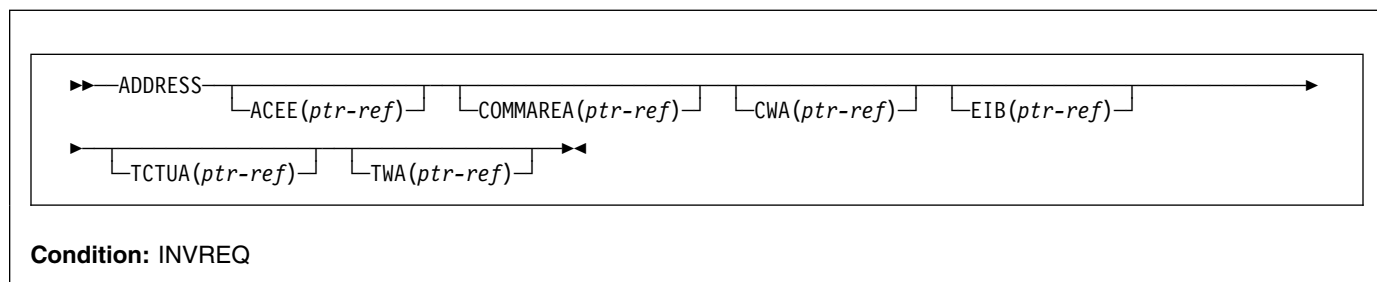
Examples

The following example shows how to terminate a task abnormally:

```
EXEC CICS ABEND ABCODE('BCDE')
```


ADDRESS

Obtain access to CICS storage areas.



Note for dynamic transaction routing

Using ADDRESS with CWA could create inter-transaction affinities that adversely affect the use of dynamic transaction routing. See the *CICS Application Programming Guide* for more information about transaction affinities.

Description

ADDRESS accesses the following areas:

- The access control environment element (ACEE)
- The communication area available to the invoked program (COMMAREA)
- The common work area (CWA)
- The EXEC interface block (EIB)
- The terminal control table user area (TCTUA)
- The transaction work area (TWA)

In assembler language, no more than four options may be specified in one ADDRESS command.

Options

ACEE(ptr-ref)

returns a pointer to the access control environment element, the control block that is generated by an external security manager (ESM) when the user signs on. If the user is not signed on, the address of the CICS DFLTUSER's ACEE is returned. If an ACEE does not exist, CICS sets the pointer reference to the null value, X'FF000000'.

Note: Take care when addressing an ACEE in a server program invoked by a distributed program link. The ACEE address returned depends on the link security and may not be the same as the address of the user signed on at the local system.

COMMAREA(ptr-ref)

returns a pointer reference, set to the address of the communication area (COMMAREA) available to the currently executing program. COMMAREA is used to

pass information between application programs. If the COMMAREA does not exist, the pointer reference is set to the null value, X'FF000000'.

In C, you must use ADDRESS COMMAREA to get the address of the communication area, because this is not passed as an argument to a C main function.

CWA(ptr-ref)

returns a pointer reference, set to the address of the common work area (CWA). This area makes information available to applications running in a single CICS system. If a CWA does not exist, CICS sets the pointer reference to the null value, X'FF000000'.

EIB(ptr-ref)

returns a pointer reference set to the address of the EXEC interface block (EIB). You must use this option to get addressability to the EIB in application routines other than the first invoked by CICS (where addressability to the EIB is provided automatically). If the application program is translated with SYSEIB in the XOPTS parameter list, this option returns the address of the system EIB.

If TASKDATALOC(ANY) is defined on the transaction definition, the address of the data may be above or below the 16MB line.

If TASKDATALOC(BELOW) is defined on the transaction definition, and the data resides above the 16MB line, the data is copied below the 16MB line, and the address of this copy is returned.

C functions must use ADDRESS EIB to get the address of the EXEC interface block, because this address is not passed as an argument to a C main function. You must code an ADDRESS EIB statement at the beginning of each application if you want access to the EIB, or if you

ADDRESS

are using a command that includes the RESP or RESP2 option.

TCTUA(*ptr-ref*)

returns a pointer reference, set to the address of the terminal control table user area (TCTUA) for the principal facility, not that for any alternate facility that may have been allocated. This area is used for passing information between application programs, but only if the same terminal is associated with the application programs involved. If a TCTUA does not exist, the pointer reference is set to the null value, X'FF000000'.

TWA(*ptr-ref*)

returns a pointer reference, set to the address of the transaction work area (TWA). This area is used for passing information between application programs, but only if they are in the same task. If a TWA does not exist, the pointer reference is set to the null value, X'FF000000'.

If TASKDATALOC(ANY) is defined on the transaction definition, the address of the data may be above or below the 16MB line.

If TASKDATALOC(BELOW) is defined on the transaction definition, and the data resides above the 16MB line, the data is copied below the 16MB line, and the address of this copy is returned.

Conditions

INVREQ

RESP2 values:

200 The TCTUA option is specified on an ADDRESS command issued in a DPL server program.

Default action: terminate the task abnormally.

ADDRESS SET

Set the address of a structure or pointer.

```

  ADDRESS SET(data-area) USING(ptr-ref)
  ADDRESS SET(ptr-ref) USING(data-area)
  
```

Description

The value from the USING option is used to set the reference in the SET option.

Options

SET(*data-area/ptr-ref*)
sets a pointer reference.

USING(*ptr-ref/data-area*)
supplies a pointer value.

Examples

COBOL example of ADDRESS SET

To set the address of a structure to a known pointer value:

```
EXEC CICS ADDRESS SET(address of struc)
        USING(ptr)
```

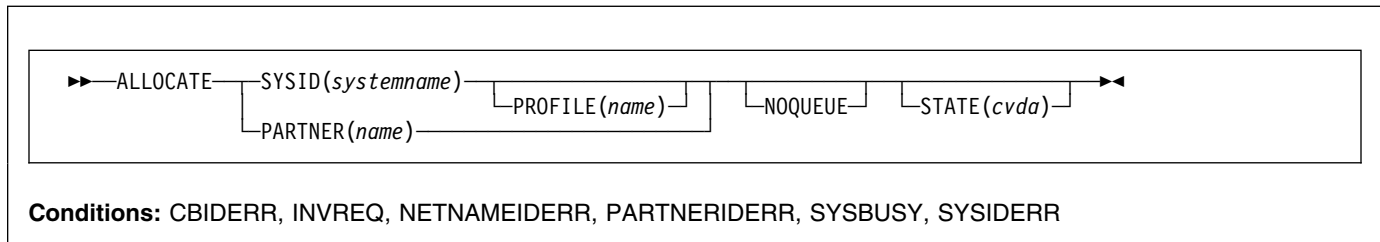
To set a pointer variable to the address of a structure:

```
EXEC CICS ADDRESS SET(ptr)
        USING(address of struc01)
```

ALLOCATE (APPC)

ALLOCATE (APPC)

Acquire a session to a remote APPC logical unit for use by an APPC mapped conversation.



Conditions: CBIDERR, INVREQ, NETNAMEIDERR, PARTNERIDERR, SYSBUSY, SYSIDERR

Description

ALLOCATE acquires the session and optionally selects a set of session-processing options. CICS makes one of the sessions associated with the named system available to the application program.

CICS returns, in EIBRSRCE in the EIB, the 4-byte CONVID (conversation identifier) that the application program uses in all subsequent commands that relate to the conversation.

If the session requested is not available, the application is suspended until the session does become available. In such a case, the suspension of the application can be prevented by specifying either the NOQUEUE or the NOSUSPEND option. NOSUSPEND is still supported as an equivalent for NOQUEUE, but NOQUEUE is the preferred keyword.

A session is available for allocation only if it is *all* of the following:

- A contention winner
- Already bound
- Not already allocated

Options

NOQUEUE

overrides the default action when a SYSBUSY condition arises. This indicates that the session requested is not immediately available. The default action is to suspend application execution until the session is available. NOQUEUE inhibits this waiting; control returns immediately to the application program instruction following the command.

Note, however, that if a HANDLE CONDITION for SYSBUSY is active when the command is executed, this also overrides the default action, and control is passed to the user label supplied in the HANDLE CONDITION. This takes precedence over the NOQUEUE option but is, of course, negated by either NOHANDLE or RESP.

For a single session APPC ALLOCATE request, issued from the **loser** half session, the NOQUEUE option

always causes SYSBUSY to be returned rather than allowing the request to bid for the session.

PARTNER(name)

specifies the name (8 characters) of a set of definitions that include the names of a remote LU (NETNAME) and a communication profile to be used on the allocated session. You can use this option as an alternative to specifying SYSID and PROFILE explicitly.

PROFILE(name)

specifies the name (1–8 characters) of a set of session-processing options that are to be used during the execution of mapped commands for the session specified in the SYSID option. If you specify SYSID and omit PROFILE, a default profile (DFHCICSA) is selected.

STATE(cvda)

gets the state of the current conversation. The cvda value returned by CICS is ALLOCATED.

SYSID(systemname)

specifies the name (1–4 characters) by which the remote APPC LU is known to this CICS. This option requests that one of the sessions to the named system is to be allocated.

Conditions

CBIDERR

occurs if the requested PROFILE cannot be found.
Default action: terminate the task abnormally.

INVREQ

occurs if the ALLOCATE command is not valid for the device to which it is directed.
Default action: terminate the task abnormally.

NETNAMEIDERR

occurs if the name specified in the NETNAME parameter of the RDO definition for the PARTNER specified on the allocate command is invalid.
Default action: terminate the task abnormally.

PARTNERIDERR

occurs if the name specified in the PARTNER option is not recognized by CICS.

Default action: terminate the task abnormally.

SYSBUSY

occurs for one of the following reasons:

- The request for a session cannot be serviced immediately. This is only possible if the NOQUEUE option is set, or a HANDLE CONDITION for SYSBUSY is active.
- The ALLOCATE command is issued when persistent session recovery is still in process and the sessions needed to satisfy the command are not yet recovered.

Default action: ignore the condition.

SYSIDERR

occurs if CICS is unable to provide the application program with a suitable session, for one of the following reasons:

- The name specified in the SYSID option is not recognized by CICS.
- The mode name derived from the PROFILE option is not one of the mode names defined for the APPC system entry.
- All the sessions in the group specified by SYSID and mode name are out of service, or all sessions are out of service.
- The AID (automatic initiate descriptor) representing your ALLOCATE has been canceled.

Default action: terminate the task abnormally.

ALLOCATE (LUTYPE6.1)

ALLOCATE (LUTYPE6.1)

Acquire a session to a remote LUTYPE6.1 logical unit.

```
▶▶ ALLOCATE [SESSION(name) [SYSID(systemname)] [PROFILE(name)] [NOQUEUE] ▶▶
```

Conditions: CBIDERR, EOC, INVREQ, SESSBUSY, SESSIONERR, SYSBUSY, SYSIDERR

Description

ALLOCATE acquires an alternate facility and optionally selects a set of session-processing options. If SYSID is specified, CICS makes available to the application program one of the sessions associated with the named system. The name of this session can be obtained from EIBRSRCE in the EIB. If SESSION is specified, CICS makes the named session available.

If the session requested is not available, the application is suspended until the session does become available. In such a case, the suspension of the application can be prevented by specifying either the NOQUEUE or the NOSUSPEND option. NOSUSPEND is still supported as an equivalent for NOQUEUE, but NOQUEUE is the preferred keyword.

Options

NOQUEUE

overrides the default action when a SESSBUSY or SYSBUSY condition arises. These conditions indicate that the session requested is not immediately available. The default action is to suspend application execution until the session is available. NOQUEUE inhibits this waiting; control returns immediately to the application program instruction following the command.

Note, however, that if a HANDLE CONDITION for SESSBUSY or SYSBUSY is active when the command is executed, this also overrides the default action, and control is passed to the user label supplied in the HANDLE CONDITION. This takes precedence over the NOQUEUE option but is, of course, negated by either NOHANDLE or RESP.

PROFILE(*name*)

specifies the name (1–8 characters) of a set of session-processing options that are to be used during execution of terminal control commands for the session specified in the SYSID or SESSION options. If the PROFILE option is omitted, a default profile (DFHCICSA) is selected.

SESSION(*name*)

specifies the symbolic identifier (1–4 characters) of a session TCTTE. This option specifies the alternate facility to be used.

SYSID(*systemname*)

specifies the name (1–4 characters) of a system TCTSE. This option specifies that one of the sessions to the named system is to be allocated.

Conditions

CBIDERR

occurs if the requested PROFILE cannot be found.

Default action: terminate the task abnormally.

EOC

occurs when a request/response unit (RU) is received with the end-of-chain indicator set. Field EIBEOC also contains this indicator.

Default action: ignore the condition.

INVREQ

occurs when the specified session is already allocated to this task, or the session is an APPC session.

Default action: terminate the task abnormally.

SESSBUSY

occurs if the request for the specified session cannot be serviced immediately. This is only possible if the NOQUEUE option is set, or a HANDLE CONDITION for SESSBUSY is active.

Default action: ignore the condition.

SESSIONERR

occurs if the name specified in the SESSION option is not that of an LUTYPE6.1 session TCTTE, or if the session cannot be allocated because it is out of service.

Default action: terminate the task abnormally.

SYSBUSY

occurs for one of the following reasons:

- The request for a session cannot be serviced immediately. This is only possible if the NOQUEUE

option is set, or a HANDLE CONDITION for SYSBUSY is active.

- The ALLOCATE command is issued when persistent session recovery is still in process and the sessions needed to satisfy the command are not yet recovered.

Default action: ignore the condition.

SYSIDERR

occurs if CICS is unable to provide the application program with a suitable session, for one of the following reasons:

- The name specified in the SYSID option is not recognized by CICS.
- All sessions are out of service.
- The AID (automatic initiate descriptor) representing your ALLOCATE has been canceled.

Default action: terminate the task abnormally.

ALLOCATE (MRO)

ALLOCATE (MRO)

Acquire an MRO session.

```
▶▶—ALLOCATE—SYSID(systemname)—┐NOQUEUE┐┐STATE(cvda)┐▶▶
```

Conditions: INVREQ, SYSBUSY, SYSIDERR

Description

ALLOCATE acquires an alternate facility. CICS makes available to the application program one of the sessions associated with the system named in the SYSID option. The name of this session can be obtained from EIBRSRCE in the EIB.

(If the PROFILE option is specified, it is ignored on an MRO session. This is useful if you want to code a command that can be used on either an LUTYPE6.1 or MRO session.)

If the session requested is not available, the application is suspended until the session does become available. In such a case, the suspension of the application can be prevented by specifying the NOQUEUE option.

For more information about MRO and IRC, see the *CICS Intercommunication Guide*.

Options

NOQUEUE

overrides the default action when a SYSBUSY condition arises. This condition indicates that the session requested is not immediately available. The default action is to suspend application execution until the session is available. NOQUEUE inhibits this waiting; control returns immediately to the application program instruction following the command.

Note, however, that if a HANDLE CONDITION for SYSBUSY is active when the command is executed, this also overrides the default action, and control is passed to the user label supplied in the HANDLE CONDITION. This takes precedence over the NOQUEUE option but is, of course, negated by either NOHANDLE or RESP.

STATE(*cvda*)

gets the state of the current conversation. The *cvda* value returned by CICS is ALLOCATED.

SYSID(*systemname*)

specifies the name (1–4 characters) of a system TCTSE. This option specifies that one of the sessions to the named system is to be allocated.

Conditions

INVREQ

occurs if an incorrect command has been issued for the LU or terminal in use.

Default action: terminate the task abnormally.

SYSBUSY

occurs if the request for a session cannot be serviced immediately. This is only possible if the NOQUEUE option is set, or a HANDLE CONDITION for SYSBUSY is active.

Default action: ignore the condition.

SYSIDERR

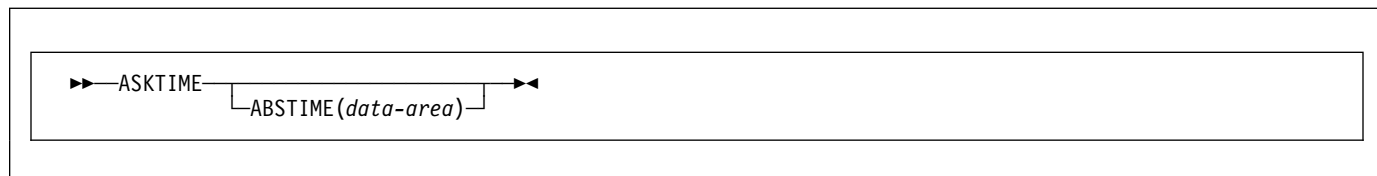
occurs if CICS is unable to provide the application program with a suitable session, for one of the following reasons:

- The name specified in the SYSID option is not recognized by CICS.
- All sessions are out of service.
- The AID (automatic initiate descriptor) representing your ALLOCATE has been canceled.

Default action: terminate the task abnormally.

ASKTIME

Request current date and time of day.



Description

ASKTIME updates the date (EIBDATE) and CICS time-of-day clock (EIBTIME) fields in the EIB. These two fields initially contain the date and time when the task started.

In response to an ASKTIME command, CICS issues an VSE STCK macro and modifies this by a local time difference. Thus, if your VSE TOD (hardware) clock is set to, say, GMT, and the local time is defined as British Summer Time (BST), it is BST that is stored in the EIBTIME field.

Refer to Appendix A, “EXEC interface block” on page 373 for details of the EIB.

Options

ABSTIME(*data-area*)

specifies the data area for the time, in packed decimal, since 00:00 on 1 January 1900 (in milliseconds rounded to the nearest hundredth of a second).

You can use FORMATTIME to change the data into other familiar formats.

Examples

For example, after execution of:

```
EXEC CICS ASKTIME ABSTIME(utime)
```

“utime” contains a value similar in format to 002837962864820.

The format of “data-area” is:

```
COBOL: PIC S9(15) COMP-3
C      char data_area[8];
PL/I:  FIXED DEC(15)
ASM:   PL8
```

ASSIGN

Request values from outside the application program's local environment.

ASSIGN	
ABCODE (data-area)	MAPWIDTH (data-area)
ABDUMP (data-area)	MSRCONTROL (data-area)
ABPROGRAM (data-area)	NATLANGINUSE (data-area)
ALTSCRNHT (data-area)	NETNAME (data-area)
ALTSCRNWD (data-area)	NEXTTRANSID (data-area)
APLKYBD (data-area)	NUMTAB (data-area)
APLTEXT (data-area)	OPCLASS (data-area)
APPLID (data-area)	OPERKEYS (data-area)
ASRAINTRPT (data-area)	OPID (data-area)
ASRAKEY (cvda)	OPSECURITY (data-area)
ASRAPSW (data-area)	ORGABCODE (data-area)
ASRAREGS (data-area)	OUTLINE (data-area)
ASRASTG (cvda)	PAGENUM (data-area)
BTRANS (data-area)	PARTNPAGE (data-area)
CMDSEC (data-area)	PARTNS (data-area)
COLOR (data-area)	PARTNSET (data-area)
CWALENG (data-area)	PRINSYSID (data-area)
DEFSCRNHT (data-area)	PROGRAM (data-area)
DEFSCRNWD (data-area)	PS (data-area)
DELIMITER (data-area)	QNAME (data-area)
DESTCOUNT (data-area)	RESSEC (data-area)
DESTID (data-area)	RESTART (data-area)
DESTIDLENG (data-area)	RETURNPROG (data-area)
DSSCS (data-area)	SCRNHT (data-area)
DS3270 (data-area)	SCRNWD (data-area)
EWASUPP (data-area)	SIGDATA (data-area)
EXTDS (data-area)	SOSI (data-area)
FACILITY (data-area)	STARTCODE (data-area)
FCI (data-area)	STATIONID (data-area)
GCHARS (data-area)	SYSID (data-area)
GCODES (data-area)	TASKPRIORITY (data-area)
GMMI (data-area)	TCTUALENG (data-area)
HIGHLIGHT (data-area)	TELLERID (data-area)
INITPARM (data-area)	TERMCODE (data-area)
INITPARMLEN (data-area)	TERMPRIORITY (data-area)
INPARTN (data-area)	TEXTKYBD (data-area)
INVOKINGPROG (data-area)	TEXTPRINT (data-area)
KATAKANA (data-area)	TRANPRIORITY (data-area)
LANGINUSE (data-area)	TWALENG (data-area)
LDCNMEN (data-area)	UNATTEND (data-area)
LDCNUM (data-area)	USERID (data-area)
MAPCOLUMN (data-area)	USERNAME (data-area)
MAPHEIGHT (data-area)	USERPRIORITY (data-area)
MAPLINE (data-area)	VALIDATION (data-area)

Condition: INVREQ

Description

ASSIGN gets values from outside the local environment of the application program. The data obtained depends on the specified options. Up to sixteen options can be specified in one ASSIGN command.

Where any of the following options apply to terminals or

terminal-related data, the reference is always to the principal facility.

If the principal facility is a remote terminal, the data returned is obtained from the local copy of the information; the request is not routed to the system to which the remote terminal is attached.

Transaction routing is as far as possible transparent to the ASSIGN command. In general, the values returned are the same whether the transaction is local or remote.

However, if the application-owning region (AOR) and the terminal-owning region (TOR) have different security requirements, security-related options *must* return different (but usually equivalent) values to the AOR and TOR.

For more details on these options, see the *CICS Intercommunication Guide*.

Options

Note: CICS Transaction Server for VSE/ESA, unlike CICS for VSE/ESA, does not support the option ASRASPC. If a program with this option is translated on VSE/ESA a warning message DFH7275 is produced and the option is ignored.

ABCODE(*data-area*)

returns a 4-character current abend code (abend codes are documented in the *VSE/ESA Messages and Codes Volume 3* manual). If an abend has not occurred, the variable is set to blanks.

ABDUMP(*data-area*)

returns a 1-byte value. X'FF' indicates that a dump has been produced and that ABCODE contains an abend code. X'00' indicates either that no dump has been produced, or that ABCODE contains blanks.

ABPROGRAM(*data-area*)

returns an 8-character name of the failing program for the latest abend.

If the abend originally occurred in a DPL server program running in a remote system, ABPROGRAM returns the DPL server program name.

This field is set to binary zeros if it is not possible to determine the failing program at the time of the abend.

ALTSCRNHT(*data-area*)

returns the alternate screen height defined for the terminal as a halfword binary variable. If the task is not initiated from a terminal, INVREQ occurs.

ALTSCRNWD(*data-area*)

returns the alternate screen width defined for the terminal as a halfword binary variable. If the task is not initiated from a terminal, INVREQ occurs.

APLKYBD(*data-area*)

returns a 1-byte indicator showing whether the terminal keyboard has the APL keyboard feature. X'FF' indicates "yes". X'00' indicates "no". If the task is not initiated from a terminal, INVREQ occurs.

APLTEXT(*data-area*)

returns a 1-byte indicator showing whether the terminal keyboard has the APL text feature. X'FF' indicates "yes". X'00' indicates "no". If the task is not initiated from a terminal, INVREQ occurs.

APPLID(*data-area*)

returns an 8-character applid of the CICS system owning the transaction.

If your system is using XRF (that is, XRF=YES has been specified in the system initialization parameters), the value returned is the *generic* applid (that is, the applid that identifies the active and alternate CICS systems). An application program is unaffected by a takeover from the active to the alternate.

ASRAINTRPT(*data-area*)

returns an 8-character program status word (PSW) containing interrupt information at the point when the latest abend with a code of ASRA, ASRB, or AICA occurred.

The field contains binary zeros if no ASRA, ASRB, or AICA abend occurred during the execution of the issuing transaction, or if the abend originally occurred in a remote DPL server program.

ASRAKEY(*cvda*)

returns the execution key at the time of the last ASRA, ASRB, AICA, or AEYD, abend, if any. CVDA values are:

CICSEXECKEY

is returned if the task was executing in CICS-key at the time of the last ASRA, ASRB, AICA, or AEYD abend. Note that all programs execute in CICS key if CICS subsystem storage protection is not active.

USEREXECKEY

is returned if the task was executing in user-key at the time of the last ASRA, ASRB, AICA, or AEYD abend.

NONCICS

is returned if the execution key at the time of the last abend was not one of the CICS keys; that is, not key 8 or key 9.

NOTAPPLIC

is returned if there has not been an ASRA, ASRB, AICA, or AEYD abend.

ASRAPSW(*data-area*)

returns an 8-character program status word (PSW) at the point when the latest abend with a code of ASRA, ASRB, ASRD, or AICA occurred.

The field contains binary zeros if no ASRA, ASRB, or AICA abend occurred during the execution of the issuing transaction, or if the abend originally occurred in a remote DPL server program.

ASRAREGS(*data-area*)

returns the contents of general registers 0–15 at the point when the latest ASRA, ASRB, ASRD, or AICA abend occurred.

The contents of the registers are returned in the data area (64 bytes long) in the order 0, 1, ..., 14, 15.

ASSIGN

Note that the data area is set to binary zeros if no ASRA, ASRB, ASRD, or AICA abend occurred during the execution of the issuing transaction or the abend originally occurred in a remote DPL server program.

ASRASTG(*cvda*)

returns the type of storage being addressed at the time of the last ASRA or AEYD, abend, if any. The CVDA values are:

CICS

is returned if the storage being addressed is CICS-key storage. This can be in one of the CICS dynamic storage areas (CDSA or ECDSA), or, in one of the read-only dynamic storage areas (RDSA or ERDSA) when CICS is running with the NOPROTECT option on the RENTPGM system initialization parameter or when storage protection is not active.

USER

is returned if the storage being addressed is user-key storage in one of the user dynamic storage areas (UDSA or EUDSA).

READONLY

is returned if the storage being addressed is read-only storage in one of the read-only dynamic storage areas (RDSA or ERDSA) when CICS is running with the PROTECT option on the RENTPGM system initialization parameter.

NOTAPPLIC

is returned if:

- There is no ASRA or AEYD abend found for this task.
- The affected storage in an abend is not managed by CICS.
- The ASRA abend is not caused by an 0C4 abend.
- An ASRB or AICA abend has occurred since the last ASRA or AEYD abend.

BTRANS(*data-area*)

returns a 1-byte indicator showing whether the terminal is defined as having the background transparency capability (X'FF') or not (X'00'). If the task is not initiated from a terminal, INVREQ occurs.

CMDSEC(*data-area*)

returns a 1-byte indicator showing whether command security checking has been defined for the current task. (X for "yes", blank for "no".)

COLOR(*data-area*)

returns a 1-byte indicator showing whether the terminal is defined as having the extended color capability (X'FF') or not (X'00'). If the task is not initiated from a terminal, INVREQ occurs.

CWALENG(*data-area*)

returns a halfword binary field indicating the length of the CWA. If no CWA exists, a zero length is returned.

DEFSCRNHT(*data-area*)

returns a halfword binary variable that contains the default screen height defined for the terminal. If the task is not initiated from a terminal, INVREQ occurs.

DEFSCRNWD(*data-area*)

returns a halfword binary variable that contains the default screen width defined for the terminal. If the task is not initiated from a terminal, INVREQ occurs.

DELIMITER(*data-area*)

returns a 1-byte data-link control character for a 3600. This can be:

- X'80'** Input ended with end-of-text (ETX).
- X'40'** Input ended with end-of-block (ETB).
- X'20'** Input ended with inter-record separator (IRS).
- X'10'** Input ended with start of header (SOH).
- X'08'** Transparent input.

If the task is not initiated from a terminal, INVREQ occurs.

DESTCOUNT(*data-area*)

returns a halfword binary field. This option has two uses:

1. Following a BMS ROUTE command, it shows that the value required is the number of different terminal types in the route list, and hence the number of overflow control areas that may be required.
2. Within BMS overflow processing, it shows that the value required is the relative overflow control number of the destination that has encountered overflow. If this option is specified when overflow processing is not in effect, the value obtained is meaningless. If no BMS commands have been issued, INVREQ occurs.

DESTID(*data-area*)

returns an 8-byte identifier of the outboard destination, padded with blanks on the right to eight characters. If this option is specified before a batch data interchange command has been issued in the task, INVREQ occurs.

DESTIDLENG(*data-area*)

returns a halfword binary length of the destination identifier obtained by DESTID. If this option is specified before a batch data interchange command has been issued in the task, INVREQ occurs.

DSSCS(*data-area*)

returns a 1-byte indicator showing whether the principal facility is a basic SCS data stream device. (X'FF' for "yes", or X'00' for "no".)

If the task is not initiated from a terminal, INVREQ occurs.

DS3270(*data-area*)

returns a 1-byte indicator showing whether the principal facility is a 3270 data stream device. (X'FF' for “yes”, or X'00' for “no”.)

If the task is not initiated from a terminal, INVREQ occurs.

EWASUPP(*data-area*)

returns a 1-byte indicator showing whether Erase Write Alternative is supported. (X'FF' for “yes”, X'00' for “no”.)

If the task is not initiated from a terminal, INVREQ occurs.

EXTDS(*data-area*)

returns a 1-byte indicator showing whether the terminal accepts the 3270 extended data stream, (X'FF') or not (X'00'). Extended data stream capability is required for a terminal that supports the query feature, color, extended highlighting, programmed symbols or validation. A terminal that accepts the query structured field command also has this indicator set. If extended data stream is on, the device supports the write structured field COMMAND and Outbound Query Structured field.

(For guidance information about query structured fields, see the *CICS 3270 Data Stream Device Guide*.)

If the task is not initiated from a terminal, INVREQ occurs.

FACILITY(*data-area*)

returns a 4-byte identifier of the principal facility that initiated the transaction issuing this command. If this option is specified, and there is no allocated facility, INVREQ occurs.

Note: You can use the QNAME option (described on page 25) to get the name of the transient data intrapartition queue if the transaction was initiated by expiry of a transient data trigger level.

FCI(*data-area*)

returns a 1-byte facility control indicator, see “ASSIGN FCI” on page 381. This indicates the type of facility associated with the transaction; for example, X'01' indicates a terminal or logical unit. The obtained value is always returned.

GCHARS(*data-area*)

returns a halfword binary graphic character set global identifier (the GCSGID). The value is a number in the range 1 through 65534 representing the set of graphic characters that can be input or output at the terminal. If the task is not initiated from a terminal, INVREQ occurs.

GCODES(*data-area*)

returns a halfword binary code page global identifier (the CPGID). The value is a number in the range 1 through 65534 representing the EBCDIC or ASCII code page defining the code points for the characters that can be

input or output at the terminal. If the task is not initiated from a terminal, INVREQ occurs.

GMMI(*data-area*)

returns a 1-byte indicator showing whether a “good morning” message applies to the terminal associated with the running transaction. (X'FF' for “yes”, or X'00' for “no”.) If this option is specified and the current task is not associated with a terminal, the INVREQ condition occurs.

HIGHLIGHT(*data-area*)

returns a 1-byte indicator showing whether the terminal is defined as having the extended highlight capability (X'FF') or not (X'00'). If the task is not initiated from a terminal, INVREQ occurs.

INITPARM(*data-area*)

returns the 60-character data-area containing any initialization parameters specified for the program in the INITPARM system initialization parameter. If there are no parameters for the program, the area is filled with binary zeros. (See the *CICS System Definition Guide* for further information about the INITPARM parameter.)

INITPARMLEN(*data-area*)

returns a halfword binary length of the INITPARM. If there is no parameter for it, INITPARMLEN contains binary zeros.

INPARTN(*data-area*)

returns the 1- or 2-character name of the most recent input partition. If no map has yet been positioned, or if BMS routing is in effect, or if the task is not initiated from a terminal, INVREQ occurs.

INVOKINGPROG(*data-area*)

returns the 8-character name of the application program that used the LINK or XCTL command to link or transfer control to the current program.

If you issue the ASSIGN INVOKINGPROG command in a remote program that was invoked by a distributed program link (DPL) command, CICS returns the name of the program that issued the DPL command.

If you issue the ASSIGN INVOKINGPROG command in an application program at the highest level, CICS returns eight blanks.

If you issue the ASSIGN INVOKINGPROG command in a user-replaceable program or a program list table program, CICS returns eight blanks.

If you issue the ASSIGN INVOKINGPROG command from a global user exit, task-related exit, or application program linked to from such an exit, CICS returns the name of the most recent invoking program that was not a global user exit or task-related user exit.

KATAKANA(*data-area*)

returns a 1-byte indicator showing whether the principal facility supports Katakana (X'FF') or not (X'00'). If the task is not initiated from a terminal, INVREQ occurs.

ASSIGN

LANGINUSE(*data-area*)

returns a 3-byte mnemonic code showing the language in use. The 3-byte mnemonic has a 1:1 correspondence with the 1-byte NATLANGINUSE option. See Appendix I, "National language codes" on page 403 for possible values of the code.

LDCMNEM(*data-area*)

returns a 2-byte logical device code (LDC) mnemonic of the destination that has encountered overflow. If this option is specified when overflow processing is not in effect, the value obtained not significant. If no BMS commands have been issued, INVREQ occurs.

LDCNUM(*data-area*)

returns a 1-byte LDC numeric value of the destination that has encountered overflow. This indicates the type of the LDC, such as printer or console. If this option is specified when overflow processing is not in effect, the value obtained is not significant.

MAPCOLUMN(*data-area*)

returns a halfword binary number of the column on the display containing the origin of the most recently positioned map. If no map has yet been positioned, or if BMS routing is in effect, or if the task is not initiated from a terminal, INVREQ occurs.

MAPHEIGHT(*data-area*)

returns a halfword binary height of the most recently positioned map. If no map has yet been positioned, or if BMS routing is in effect, or if the task is not initiated from a terminal, INVREQ occurs.

MAPLINE(*data-area*)

returns a halfword binary number of the line on the display containing the origin of the most recently positioned map. If no map has yet been positioned, or if BMS routing is in effect, or if the task is not initiated from a terminal, INVREQ occurs.

MAPWIDTH(*data-area*)

returns a halfword binary width of the most recently positioned map. If no map has yet been positioned, or if BMS routing is in effect, or if the task is not initiated from a terminal, INVREQ occurs.

MSRCONTROL(*data-area*)

returns a 1-byte indicator showing whether the terminal supports magnetic slot reader (MSR) control (X'FF') or not (X'00'). If the task is not initiated from a terminal, INVREQ occurs.

NATLANGINUSE(*data-area*)

returns a 1-byte mnemonic code showing the national language in use. See Appendix I, "National language codes" on page 403 for possible values of the code. (NATLANGINUSE does not show the system default language as specified on the NATLANG system initialization parameter.)

NETNAME(*data-area*)

returns the 8-character name of the logical unit in the VTAM® network. If the task is not initiated from a terminal, INVREQ occurs. If the principal facility is not a local terminal, CICS no longer returns a null string but the netname of the remote terminal.

NEXTTRANSID(*data-area*)

returns the 4-character next transaction identifier as set by SET NEXTTRANSID or RETURN TRANSID. It returns blanks if there are no more transactions.

NUMTAB(*data-area*)

returns a 1-byte number of the tabs required to position the print element in the correct passbook area of the 2980. If the task is not initiated from a terminal, INVREQ occurs.

OPCLASS(*data-area*)

returns, in a 24-bit string, the operator class used by BMS for routing terminal messages, as defined in the CICS segment of the External Security Manager (ESM).

OPERKEYS(*data-area*)

is accepted for compatibility with previous releases. If specified, a 64-bit null string is returned.

OPID(*data-area*)

returns the 3-character operator identification. This is used by BMS for routing terminal messages, as defined in the CICS segment of the External Security Manager (ESM).

If the task is initiated from a remote terminal, the OPID returned by this command is not necessarily that associated with the user that is signed on at the remote terminal. If you wish to know the OPID of the signed on user, you should use the INQUIRE TERMINAL system programming command, which is described in the *CICS System Programming Reference* manual.

The OPID may also be different from that of the user currently signed on, if it has been changed with the SET TERMINAL command.

OPSECURITY(*data-area*)

is accepted for compatibility with previous releases. If specified, a 24-bit null string is returned.

ORGABCODE(*data-area*)

returns as a 4-byte original abend code in cases of repeated abends.

OUTLINE(*data-area*)

returns a 1-byte indicator showing whether the terminal is defined as having the field outlining capability (X'FF') or not (X'00'). If the task is not initiated from a terminal, INVREQ occurs.

PAGENUM(*data-area*)

returns a halfword binary current page number for the destination that has encountered an overflow. If this option is specified when overflow processing is not in

effect, the value obtained is meaningless. If no BMS commands have been issued, INVREQ occurs.

PARTNPAGE(*data-area*)

returns a 2-byte name of the partition that most recently caused page overflow. If no BMS commands have been issued, INVREQ occurs.

PARTNS(*data-area*)

returns a 1-byte indicator showing whether the terminal supports partitions (X'FF') or not (X'00'). If the task is not initiated from a terminal, INVREQ occurs.

PARTNSET(*data-area*)

returns the name (1–6 characters) of the application partition set. A blank value is returned if there is no application partition set. If the task is not initiated from a terminal, INVREQ occurs.

PRINSYSID(*data-area*)

returns the 4-character name by which the other system is known in the local system; that is, the RDO CONNECTION resource definition that defines the other system. For a single-session APPC device defined by a terminal definition, the returned value is the terminal identifier.

This only applies when the principal facility is one of the following:

- An MRO session to another CICS system
- An LU6.1 session to another CICS or IMS™ system
- An APPC session to another CICS system, or to another APPC system or device

If the principal facility is not an MRO, LU6.1, or APPC session, or if the task has no principal facility, INVREQ occurs.

Note: Special considerations apply generally when transaction routing. In particular an ASSIGN PRINSYSID command cannot be used in a routed transaction to find the name of the terminal-owning region. (See the *CICS Intercommunication Guide* for more information about transaction routing.)

PROGRAM(*data-area*)

returns an 8-character name of the currently running program.

PS(*data-area*)

returns a 1-byte indicator showing whether the terminal is defined as having the programmed symbols capability (X'FF') or not (X'00'). If the task is not initiated from a terminal, INVREQ occurs.

QNAME(*data-area*)

returns a 4-character name of the transient data intrapartition queue that caused this task to be initiated by reaching its trigger level. If the task is not initiated by automatic transaction initiation (ATI), INVREQ occurs.

RESSEC(*data-area*)

returns a 1-byte indicator showing whether resource security checking has been defined for the transaction running. (X for “yes”, blank for “no”.)

RESTART(*data-area*)

returns a 1-byte indicator showing whether a restart of the task (X'FF'), or a normal start of the task (X'00'), has occurred.

RETURNPROG(*data-area*)

returns the 8-character name of the program to which control is to be returned when the current program has finished executing. The values returned depend on how the current program was given control, as follows:

- If the current program was invoked by a LINK command, including a distributed program link, RETURNPROG returns the same name as INVOKINGPROG.
- If the current program was invoked by an XCTL command, RETURNPROG returns the name of the application program in the chain that last issued a LINK command.

If the program that invoked the current program with an XCTL command is at the highest level, CICS returns eight blanks.
- If the ASSIGN RETURNPROG command is issued in the program at the top level, CICS returns eight blanks.
- If the ASSIGN RETURNPROG command is issued in a user-replaceable module, or a program list table program, CICS returns eight blanks.
- If the ASSIGN RETURNPROG is issued in a global user exit, task-related exit, or application program linked to from such an exit, CICS returns the name of the program that control is returned to when all intermediate global user exit and task-related user exit programs have completed.

SCRNHT(*data-area*)

returns a halfword binary variable that contains the height of the 3270 screen defined for the current task. If the task is not initiated from a terminal, INVREQ occurs.

SCRNWD(*data-area*)

returns a halfword binary variable that contains the width of the 3270 screen defined for the current task. If the task is not initiated from a terminal, INVREQ occurs.

SIGDATA(*data-area*)

returns a 4-byte character string containing the inbound signal data received from a logical unit. If the task is not initiated from a terminal, INVREQ occurs.

SOSI(*data-area*)

returns a 1-byte indicator showing whether the terminal is defined as having the mixed EBCDIC/DBCS fields capability (X'FF') or not (X'00'). The DBCS subfields within an EBCDIC field are delimited by SO (shift-out)

ASSIGN

and SI (shift-in) characters. If the task is not initiated from a terminal, INVREQ occurs.

STARTCODE(*data-area*)

returns a 2-byte indicator showing how the transaction issuing the request was started. It can have the following values:

Code Transaction started by

- D** A distributed program link (DPL) request that did not specify the SYNCONRETURN option. The task cannot issue I/O requests against its principal facility, nor can it issue any syncpoint requests.
- DS** A distributed program link (DPL) request, as in code D, that did specify the SYNCONRETURN option. The task can issue syncpoint requests.
- QD** Transient data trigger level.
- S** START command without data.
- SD** START command with data.
- SZ** FEPI START command.
- TD** Terminal input or permanent transid.
- U** User-attached task.

STATIONID(*data-area*)

returns a 1-byte station identifier of a 2980. If the task is not initiated from a terminal, INVREQ occurs.

SYSID(*data-area*)

returns the 4-character name given to the local CICS system. This value may be specified in the SYSID option of a file control, interval control, temporary storage, or transient data command, in which case the resource to be accessed is assumed to be on the local system.

TASKPRIORITY(*data-area*)

returns a halfword binary field indicating the current priority of the issuing task (0–255). When the task is first attached, this is the sum of the user, terminal, and transaction priorities. This value can be changed during execution by a CHANGE TASK command.

TCTUALENG(*data-area*)

returns a halfword binary length of the terminal control table user area (TCTUA). If no TCTUA exists, a zero length is returned.

TELLERID(*data-area*)

returns a 1-byte teller identifier of a 2980. If the task is not initiated from a terminal, INVREQ occurs.

TERMCODE(*data-area*)

returns a 2-byte code giving the type and model number of the terminal associated with the task.

The first byte is a code identifying the terminal type, derived from the DEVICE attribute of the RDO TERMINAL resource definition (described in the *CICS Resource Definition Guide*). The second byte is a

single-character model number as specified in the TERMMODEL attribute.

The meanings of the type codes are given in Appendix B, “Codes returned by ASSIGN” on page 381.

TERMPRIORITY(*data-area*)

returns a halfword binary terminal priority (0–255).

TEXTKYBD(*data-area*)

returns a 1-byte indicator showing whether the principal facility supports TEXTKYBD. (X'FF' for “yes”, or X'00' for “no”.) If the task is not initiated from a terminal, INVREQ occurs.

TEXTPRINT(*data-area*)

returns a 1-byte indicator showing whether the principal facility supports TEXTPRINT. (X'FF' for “yes”, or X'00' for “no”.) If the task is not initiated from a terminal, INVREQ occurs.

TRANPRIORITY(*data-area*)

returns a halfword binary transaction priority (0–255).

TWALENG(*data-area*)

returns a halfword binary length of the transaction work area (TWA). If no TWA exists, a zero length is returned.

UNATTEND(*data-area*)

returns a 1-byte indicator showing whether the mode of operation of the terminal is unattended, that is to say no person is actually attending the terminal. These indicators are X'FF' for unattended and X'00' for attended. If the task is not initiated from a terminal, INVREQ occurs.

USERID(*data-area*)

returns an 8-byte userid of the signed-on user. If no user is explicitly signed on, CICS returns the default userid. Special considerations apply if you are using an intercommunication environment. See the *CICS Intercommunication Guide* for more information about the ASSIGN command for LUTYPE6.1, APPC, and MRO.

Note: If you want to test if the user is explicitly signed on, use the UNATTEND option.

USERNAME(*data-area*)

returns a 20-character name of the user obtained from the external security manager (ESM).

USERPRIORITY(*data-area*)

returns a halfword binary operator priority (0–255).

VALIDATION(*data-area*)

returns a 1-byte indicator showing whether the terminal is defined as having the validation capability (X'FF') or not (X'00'). Validation capability consists of the mandatory fill, mandatory enter, and trigger attributes. If the task is not initiated from a terminal, INVREQ occurs.

Conditions

INVREQ

RESP2 values:

- 2 No BMS command has yet been issued, BMS routing is in effect, or no map has yet been positioned.
- 3 No batch data interchange (BDI) command has yet been issued. See the *CICS Application Programming Guide* for BDI values.
- 4 The task is not initiated by automatic transaction initiation (ATI).
- 5 The task is not associated with a terminal; or the task has no principal facility; or the principal facility is not an MRO, LU6.1, or APPC session.
- 200 Command syntax options are not allowed in a server program invoked by a distributed program link.

Default action: terminate the task abnormally.

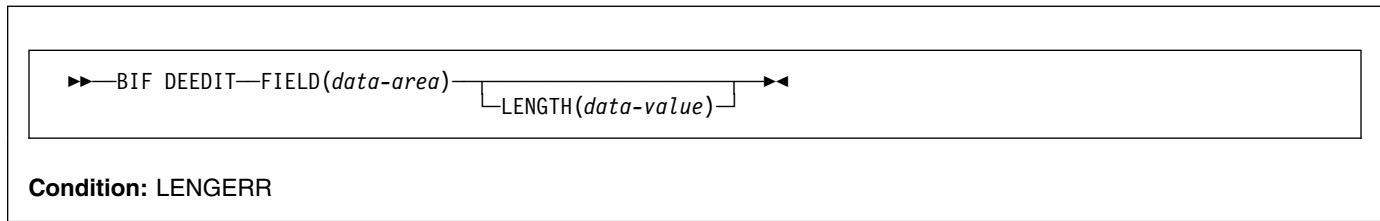
Examples

An example of RETURNPROG:

```
Program A links to program B
Program B links to program C
Program C transfers control to program D
Program D issues an ASSIGN RETURNPROG command,
and CICS returns the name of Program B.
```

BIF DEEDIT

Deediting (built-in function).



Description

BIF DEEDIT provides the built-in function DEEDIT. It specifies that alphabetic and special characters are removed from an EBCDIC data field, and the remaining digits right-aligned and padded to the left with zeros as necessary.

If the field ends with a minus sign or a carriage-return (CR), a negative zone (X'D') is placed in the rightmost (low-order) byte.

If the zone portion of the rightmost byte contains one of the characters X'A' through X'F', and the numeric portion contains one of the hexadecimal digits X'0' through X'9', the rightmost byte is returned unaltered (see the example). This permits the application program to operate on a zoned numeric field. The returned value is in the field that initially contained the unedited data.

Note that a 1-byte field is returned unaltered, no matter what the field contains.

Options

FIELD(*data-area*)
specifies the field to be edited.

LENGTH(*data-value*)
specifies the field length in bytes.

Conditions

LENGERR
occurs if the LENGTH value is less than 1.
Default action: terminate the task abnormally.

Examples

```
EXEC CICS BIF DEEDIT
      FIELD(CONTG)
      LENGTH(9)
```

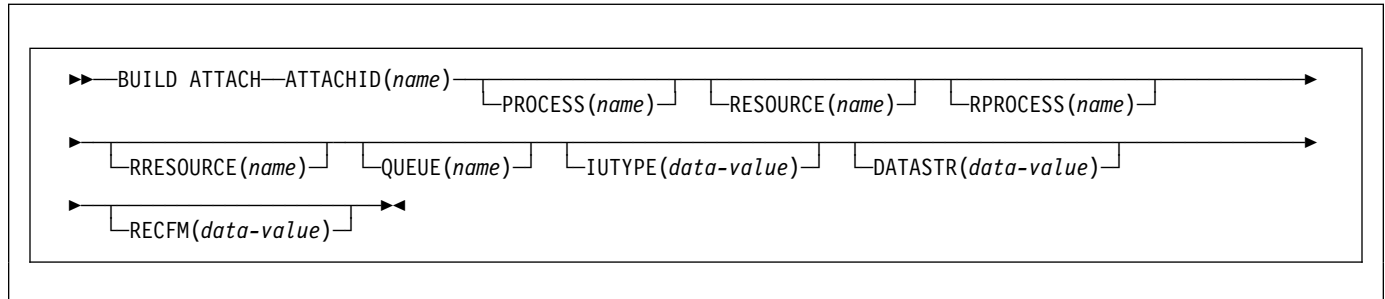
This removes all characters other than digits from CONTG, a 9-byte field, and returns the edited result in that field to the application program. Two examples of the contents of CONTG before and after execution of the command are:

Original value	Returned value
14-6704/B	00146704B
\$25.68	000002568

Note that a decimal point is an EBCDIC special character and as such is removed.

BUILD ATTACH (LUTYPE6.1)

Specify values for an LUTYPE6.1 attach header.



Description

BUILD ATTACH (LUTYPE6.1) specifies a set of values to be placed in the named attach header control block. This control block contains values that are to be sent in an LUTYPE6.1 attach FMH (Function Management Header) that is constructed by CICS, and is sent only when a SEND ATTACHID or CONVERSE ATTACHID command is executed. The specified values override existing values in the control block; unspecified values are set to default values.

Options

ATTACHID(*name*)

specifies that the set of values is to be placed in an attach header control block identified by the specified name (1–8 characters).

DATASTR(*data-value*)

corresponds to the data stream profile field, ATTDSP, in an LUTYPE6.1 attach FMH.

For communication between two CICS systems, no particular significance is attached by CICS to the data stream profile field in an attach FMH. For most CICS applications, the option can be omitted.

For details of communication between a CICS system and another subsystem, including details of structured fields and logical record management, refer to documentation supplied by the subsystem about how to use the data stream profile field in an attach FMH.

The “data-value” is a halfword binary. Only the low-order byte is used. The SNA-defined meanings of the bits are as follows:

0–7	reserved - must be set to zero
8–11	0000 - user-defined
	1111 - SCS data stream
	1110 - 3270 data stream
	1101 - structured field
	1100 - logical record management
12–15	defined by the user if bits 8–11 are set to 0000; otherwise reserved (must be set to zero)

A value of “structured field” indicates that chains begin with four bytes of data that are used to interpret the following data: overall length (2 bytes), class identifier (1 byte), and subclass identifier (1 byte). A value of “logical record management” indicates that chains can be split into separate fields by the data receiver.

If the option is omitted, a value of “user-defined” is assumed.

IUTYPE(*data-value*)

corresponds to the interchange unit field, ATTIU, in an LUTYPE6.1 attach FMH.

The “data-value” is a halfword binary. Only the low-order 7 bits are used. The SNA-defined meanings of the bits are as follows:

0–10	reserved - must be set to zero
11	0 - not end of multichain interchange unit
	1 - end of multichain interchange unit
12,13	reserved - must be set to zero
14,15	00 - multichain interchange unit
	01 - single-chain interchange unit
	10 - reserved
	11 - reserved

BUILD ATTACH (LUTYPE6.1)

If the option is omitted, values of “not end of multichain interchange unit” and “multichain interchange unit” are assumed.

PROCESS(*name*)

corresponds to the process name, ATTDPN, in an LUTYPE6.1 attach FMH.

For communication between two CICS systems, a transaction running in one system can acquire a session to the second system and can identify the transaction to be attached; in the second system, the identification is carried in the first chain of data sent across the session.

In general, the first four bytes of data identify the transaction to be attached. However an attach FMH, identifying the transaction to be attached, can be built and sent; the PROCESS option is used to specify the transaction name. (Note that the receiving CICS system uses just the first four bytes of the process name as a transaction name.)

No significance is attached by CICS to process names in attach FMHs sent in chains of data other than the first.

For communication between a CICS system and another subsystem, refer to documentation supplied by the subsystem about how to use the process name field in an attach FMH.

QUEUE(*name*)

corresponds to the queue name, ATTDQN, in an LUTYPE6.1 attach FMH.

For communication between two CICS systems, no significance is attached by CICS to the queue name in an attach FMH.

For communication between a CICS system and another subsystem, refer to documentation supplied by the subsystem about how to use the queue name field in an attach FMH.

RECFM(*data-value*)

corresponds to the deblocking algorithm field, ATTDDBA, in an LUTYPE6.1 attach FMH.

For communication between two CICS systems, no particular significance is attached by CICS to the deblocking algorithm field in an attach FMH. For most CICS applications, the option can be omitted.

The “data-value” is a halfword binary value. Only the low-order byte is used. The SNA-defined meanings of the bits are as follows:

0-7	reserved - must be set to zero
8-15	X'00' - reserved
	X'01' - variable-length variable-blocked
	X'02' - reserved
	X'03' - reserved
	X'04' - chain of RUs
	X'05' through X'FF' - reserved

If the option is omitted, a value of “chain of RUs” is assumed.

RESOURCE(*name*)

corresponds to the resource name, ATTPRN, in an LUTYPE6.1 attach FMH.

RPROCESS(*name*)

corresponds to the return-process name, ATTRDPN, in an LUTYPE6.1 attach FMH.

For communication between two CICS systems, no significance is attached by CICS to the return-process name in an attach FMH.

For communication between a CICS system and another subsystem, refer to documentation supplied by the subsystem about how to use the return-process name field in an attach FMH.

RRESOURCE(*name*)

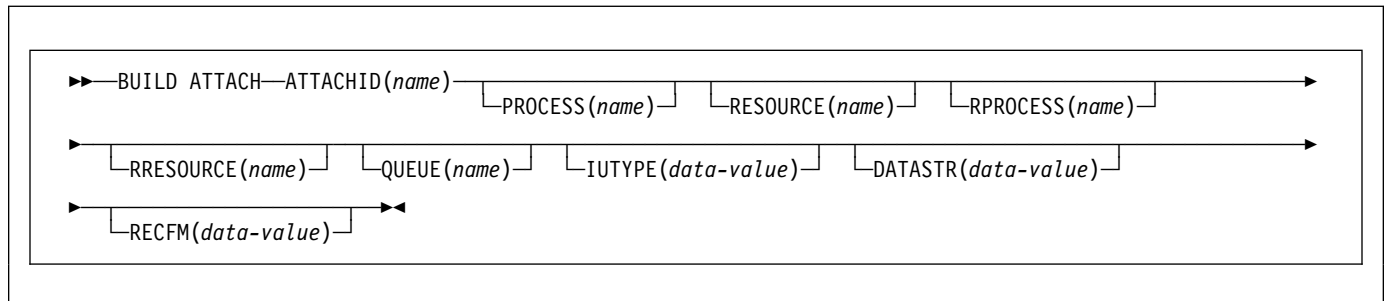
corresponds to the return-resource name, ATTRPRN, in an LUTYPE6.1 attach FMH.

For communication between two CICS systems, no significance is attached by CICS to the return-resource name in an attach FMH.

For communication between a CICS system and another subsystem, refer to documentation supplied by the subsystem about how to use the return-resource name field in an attach FMH.

BUILD ATTACH (MRO)

Specify values for an MRO attach header.



Description

BUILD ATTACH (MRO) specifies a set of values to be placed in the named attach header control block. This control block contains values that are to be sent in an MRO attach FMH (Function Management Header) that is constructed by CICS, and is sent only when a SEND ATTACHID or CONVERSE ATTACHID command is executed. The specified values override existing values in the control block; unspecified values are set to default values.

For more information about MRO and IRC, see the *CICS Intercommunication Guide*.

Options

ATTACHID(name)

specifies that the set of values is to be placed in an attach header control block identified by the specified name (1–8 characters).

DATASTR(data-value)

corresponds to the data stream profile field, ATTDSP, in an LUTYPE6.1 attach FMH.

For communication between two CICS systems, no particular significance is attached by CICS to the data stream profile field in an attach FMH. For most CICS applications, the option can be omitted.

The “data-value” is a binary halfword. Only the low-order byte is used. The SNA-defined meanings of the bits are as follows:

0–7	reserved - must be set to zero
8–11	0000 - user-defined
	1111 - SCS data stream
	1110 - 3270 data stream
	1101 - structured field
	1100 - logical record management
12–15	defined by the user if bits 8–11 are set to 0000; otherwise reserved (must be set to zero)

A value of “structured field” indicates that chains begin with four bytes of data that are used to interpret the following data; overall length (2 bytes), class identifier (1 byte), and subclass identifier (1 byte). A value of “logical record management” indicates that chains can be split into separate fields by the data receiver.

If the option is omitted, a value of “user-defined” is assumed.

IUTYPE(data-value)

corresponds to the interchange unit field, ATTIU, in an LUTYPE6.1 attach FMH.

The “data-value” is a halfword binary. Only the low-order 7 bits are used. The SNA-defined meanings of the bits are as follows:

0–10	reserved - must be set to zero
11	0 - not end of multichain interchange unit
	1 - end of multichain interchange unit
12,13	reserved - must be set to zero
14,15	00 - multichain interchange unit
	01 - single chain interchange unit
	10 - reserved
	11 - reserved

If the option is omitted, values of “not end of multichain interchange unit” and “multichain interchange unit” are assumed.

BUILD ATTACH (MRO)

PROCESS(*name*)

corresponds to the process name, ATTDPN, in an LUTYPE6.1 attach FMH.

For communication between two CICS systems, a transaction running in one system can acquire a session to the second system and can identify the transaction to be attached; in the second system the identification is carried in the first chain of data sent across the session. In general, the first four bytes of data identify the transaction to be attached. However an attach FMH, identifying the transaction to be attached, can be built and sent; the PROCESS option is used to specify the transaction name. (Note that the receiving CICS system uses just the first four bytes of the process name as a transaction name.)

No significance is attached by CICS to process names in attach FMHs sent in chains of data other than the first.

For communication between a CICS system and another subsystem, refer to documentation supplied by the subsystem about how to use the process name field in an attach FMH.

QUEUE(*name*)

corresponds to the queue name, ATTDQN, in an attach FMH.

For communication between two CICS systems, no significance is attached by CICS to the queue name in an attach FMH.

For communication between a CICS system and another subsystem, refer to documentation supplied by the subsystem about how to use the queue name field in an attach FMH.

RECFM(*data-value*)

corresponds to the deblocking algorithm field, ATTDDBA, in an LUTYPE6.1 attach FMH.

For communication between two CICS systems, no particular significance is attached by CICS to the deblocking algorithm field in an attach FMH. For most CICS applications, the option can be omitted.

The “data-value” is a halfword binary value. Only the low-order 8 bits are used. The SNA-defined meanings of the bits are as follows:

0-7	reserved - must be set to zero
8-15	X'00' - reserved
	X'01' - variable-length variable-blocked
	X'02' - reserved
	X'03' - reserved
	X'04' - chain of RUs
	X'05' to X'FF' - reserved

If the option is omitted, a value of “chain of RUs” is assumed.

RESOURCE(*name*)

corresponds to the resource-name, ATTPRN, in an LUTYPE6.1 attach FMH.

RPROCESS(*name*)

corresponds to the return-process name, ATTRDPN, in an LUTYPE6.1 attach FMH.

For communication between two CICS systems, no significance is attached by CICS to the return-process name in an attach FMH.

For communication between a CICS system and another subsystem, refer to documentation supplied by the subsystem about how to use the return-process name field in an attach FMH.

RRESOURCE(*name*)

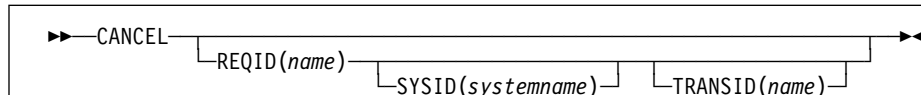
corresponds to the return-resource name, ATTRPRN, in an LUTYPE6.1 attach FMH.

For communication between two CICS systems, no significance is attached by CICS to the return-resource name in an attach FMH.

For communication between a CICS system and another subsystem, refer to documentation supplied by the subsystem about how to use the return-resource name field in an attach FMH.

CANCEL

Cancel interval control requests.



Conditions: ISCINVREQ, NOTAUTH, NOTFND, SYSIDERR

Note for dynamic transaction routing

Using CANCEL with REQID (of a POST, DELAY, or START) could create inter-transaction affinities that adversely affect the use of dynamic transaction routing. See the *CICS Application Programming Guide* for more information about transaction affinities.

Description

CANCEL cancels a previously issued DELAY, POST, or START command. If you include the SYSID option, the command is shipped to a remote system. If you omit SYSID, the TRANSID option, if present, indicates where the command is to be executed. The effect of the cancellation varies depending on the type of command being canceled, as follows:

- A DELAY command can be canceled only before it has expired, and only by a task other than the task that issued the DELAY command (which is suspended for the duration of the request). The REQID used by the suspended task must be specified. The effect of the cancellation is the same as an early expiration of the original DELAY. That is, the suspended task becomes dispatchable as though the original expiration time had been reached.
- When a POST command issued by the same task is to be canceled, no REQID need be specified. Cancellation can be requested either before or after the original request has expired. The effect of the cancellation is as if the original request had never been made.
- When a POST command issued by another task is to be canceled, the REQID of that command must be specified. The effect of the cancellation is the same as an early expiration of the original POST request. That is, the timer event control area for the other task is posted as though the original expiration time had been reached.
- When a START command is to be canceled, the REQID associated with the original command must be specified. The effect of the cancellation is as if the original command had never been issued. The cancellation is effective only before the original command has been honored.

Options

REQID(name)

specifies a name (1–8 characters), which should be unique, to identify a command. This name is used as a temporary storage identifier. The temporary storage queue thus identified must be defined as a local queue on the CICS system where the CANCEL command is processed.

This option cannot be used to cancel a POST command issued by the same task (for which, the REQID option is ignored if it is specified).

SYSID(systemname)

(remote systems only) specifies the name (1–4 characters) of the system for the CANCEL command.

TRANSID(name)

specifies the symbolic identifier (1–4 characters) of a transaction to be used to determine where the CANCEL command is to be executed, if SYSID is not specified. If the TRANSID is defined as REMOTE, the CANCEL request is function-shipped to the remote system.

Conditions

ISCINVREQ

occurs when the remote system indicates a failure that does not correspond to a known condition.

Default action: terminate the task abnormally.

NOTAUTH

occurs when a resource security check has failed on the specified TRANSID or on the TRANSID of the START command that corresponds to the request identification.

CANCEL

Default action: terminate the task abnormally.

NOTFND

occurs if the request identifier specified fails to match an unexpired interval control command.

Default action: terminate the task abnormally.

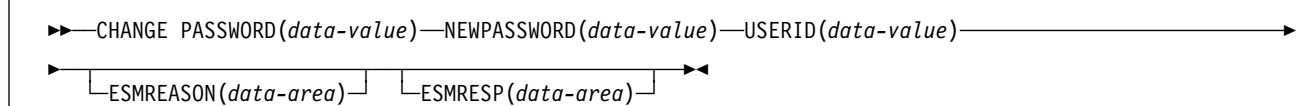
SYSIDERR

occurs when the SYSID option specifies a name that is neither the local system nor a remote system (made known to CICS by defining an RDO CONNECTION resource definition). It also occurs when the link to the remote system is closed.

Default action: terminate the task abnormally.

CHANGE PASSWORD

Change the password recorded by an external security manager (ESM) for a specified userid.



Conditions: INVREQ, NOTAUTH, USERIDERR

Description

Unlike the SIGNON command, CHANGE PASSWORD does not depend upon the principal facility, so it can be issued when the facility is an APPC session.

Attention: You should clear the password fields on the EXEC CICS commands that have a password option as soon as possible after use. This is to ensure that passwords are not revealed in system or transaction dumps.

Options

Options ESMRESP and ESMREASON return the response and reason codes, if any, from the external security manager.

ESMREASON(*data-area*)

returns the reason code, in a fullword binary field, that CICS receives from the external security manager.

ESMRESP(*data-area*)

returns the response code, in a fullword binary field, that CICS receives from the external security manager.

NEWPASSWORD(*data-value*)

specifies the new password, 8 characters, for the specified userid. The password is changed only if the current password is correctly specified.

PASSWORD(*data-value*)

specifies the current password, 8 characters, for the specified userid.

USERID(*data-value*)

specifies the userid, 8 characters, of the user whose password is being changed.

Conditions

INVREQ

RESP2 values:

- 13** There is an unknown return code in ESMRESP from the external security manager.
- 18** The CICS external security manager interface is not initialized.
- 29** The external security manager is not responding.

Default action: terminate the task abnormally.

NOTAUTH

RESP2 values:

- 2** The supplied password is wrong.
- 4** The new password is not acceptable.
- 19** The USERID is revoked.
- 22** The change password request failed during SECLABEL processing.
- 31** The user is revoked in the connection to the default group.

Default action: terminate the task abnormally.

USERIDERR

RESP2 values:

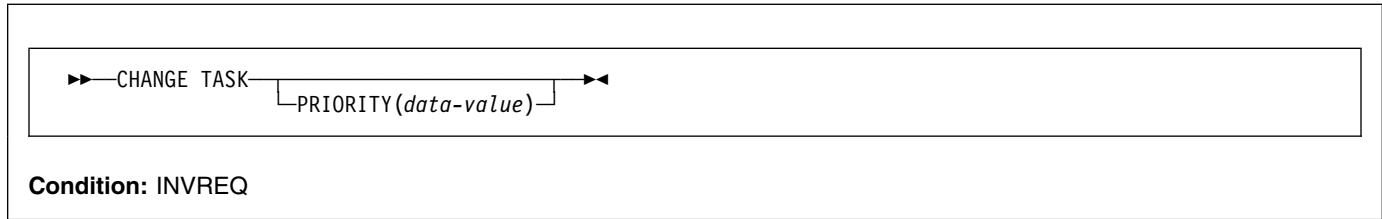
- 8** The USERID is not known to the external security manager.

Default action: terminate the task abnormally.

CHANGE TASK

CHANGE TASK

Change priority of a task.



Condition: INVREQ

Description

CHANGE TASK changes the priority of the issuing task. It has immediate effect (unlike SET TASK), because control is relinquished during execution of the command so that the current task has to be redispached. The redispach does not happen until tasks that are of higher or equal priority, and that are also ready to run, are dispatched.

If you omit the PRIORITY option, the task does not lose control and the priority remains the same. This is effectively a no-op.

Options

PRIORITY(*data-value*)

specifies a fullword binary value in the range 0–255, defining the priority of the task. You can also have a value of –1 but this does not change the priority or cause a redispach.

Conditions

INVREQ

RESP2 values:

- 1 Your PRIORITY value is outside the range -1–255.

CHKP (DL/I)

Request a basic checkpoint.

```
▶▶—CHKP—ID(data-area)—▶▶  
└('char-expr')┘
```

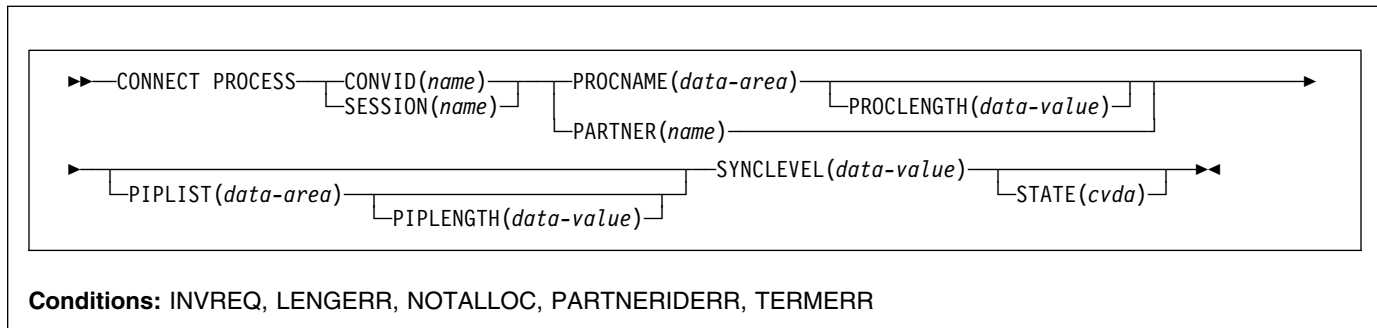
Condition:

Description

The full form of the command is CHECKPOINT. See the *DL/I DOS/VS Application Programming: High Level Programming Interfaces* manual for further details.

CONNECT PROCESS

Initiate APPC mapped conversation.



Description

CONNECT PROCESS allows an application to specify a process name and synchronization level to be passed to CICS and used when the remote partner is attached.

Options

CONVID(name)

identifies the conversation to which the command relates. The 4-character name specifies the token returned by a previously executed ALLOCATE command in EIBRSRCE in the EIB.

For compatibility with earlier releases, SESSION is accepted as a synonym for CONVID. New programs should use CONVID.

PARTNER(name)

specifies the name (8 characters) of a set of definitions that includes the name (or extended name) of a remote partner transaction (TPNAME or XTPNAME). You can use this option as an alternative to PROCNAME and PROCLENGTH.

PIPELENGTH(data-value)

specifies the total length (halfword binary value) of the specified process initialization parameter (PIP) list.

PIPLIST(data-area)

specifies the PIP data to be sent to the remote system. The PIP list consists of variable-length records, each containing a single PIP. A PIP starts with a 2-byte inclusive length field (LL), followed by a 2-byte reserved field, and then the parameter data.

PROCLENGTH(data-value)

specifies the length (as a halfword binary value in the range 1–64) of the name specified by the PROCNAME option.

PROCNAME(data-area)

specifies the partner process (that is, the transaction) to be attached in the remote system.

One byte is sufficient to identify a CICS transaction. The APPC architecture allows a range of 1–64 bytes but leaves each product free to set its own maximum. CICS complies by allowing a range of 1–64 bytes. If the remote system is CICS, this option can specify the 4-byte transaction identifier or the TPNAME value given in the relevant TRANSACTION definition. Alternatively, you can examine the full identifier by coding the user exit XZCATT.

No character checking is performed on the TPN by CICS.

For programming information about the user exit XZCATT, see the *CICS Customization Guide*.

SESSION(name)

specifies the symbolic identifier (1–4 characters) of a session TCTTE. This option specifies the alternate facility to be used.

STATE(cvda)

gets the state of the current conversation. The cvda values returned by CICS are:

ALLOCATED
 CONFREE
 CONFRECEIVE
 CONFSEND
 FREE
 PENDFREE
 PENDRECEIVE
 RECEIVE
 ROLLBACK
 SEND
 SYNCFREE
 SYNCRECEIVE
 SYNCSEND

SYNCLEVEL(*data-value*)

specifies the synchronization level (halfword binary value) for the current conversation. The possible values are:

- 0 None
- 1 Confirm
- 2 Syncpoint

Conditions**INVREQ**

RESP2 values:

200 A distributed program link server application specified the function-shipping session (its principal facility) on the CONVID option.

also occurs (RESP2 not set) in any of the following situations:

- A synchronization level other than 0, 1, or 2, has been requested in the SYNCLEVEL option.
- The command is not valid for the terminal or LU in use.
- The command has been used on a conversation that is an APPC basic conversation.

Default action: terminate the task abnormally.

LENGERR

occurs in any of the following situations:

- An out-of-range value is supplied in the PROCLENGTH option.
- The value specified in the PIPLength option is less than 0.
- The value specified in the PIPLength option exceeds the CICS implementation limit of 32 763.
- A PIPLIST length element (LL) has a value less than 4.
- The sum of the length elements (LLs) in the PIPLIST does not equal the value specified by PIPLength.

Default action: terminate the task abnormally.

NOTALLOC

occurs if the specified CONVID value does not relate to a conversation owned by the application.

Default action: terminate the task abnormally.

PARTNERIDERR

occurs if the name specified in the PARTNER option is not recognized by CICS.

Default action: terminate the task abnormally.

TERMERR

occurs for a session-related error. Any action on that conversation other than a FREE causes an ATCV abend.

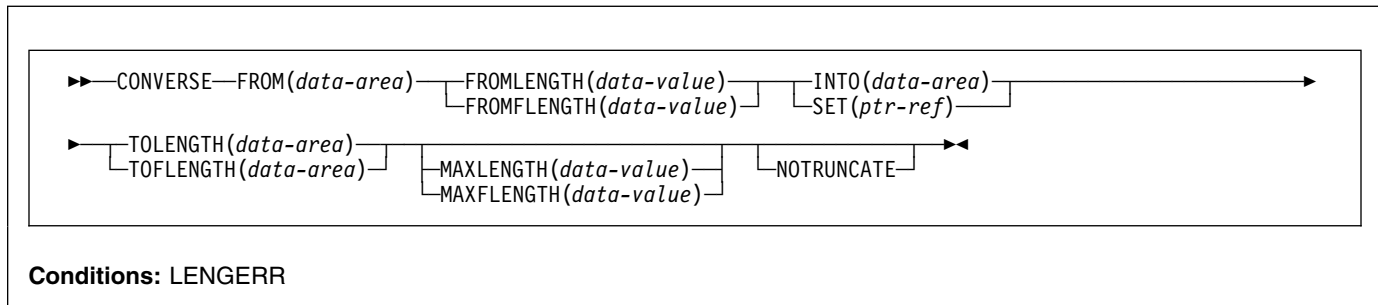
A CANCEL TASK request by a user node error program (NEP) can cause this condition if the task has an outstanding terminal control request active when the node abnormal condition program handles the session error.

Default action: terminate the task abnormally with abend code ATNI.

CONVERSE (VTAM)

CONVERSE (VTAM default)

Communicate on standard CICS terminal support.

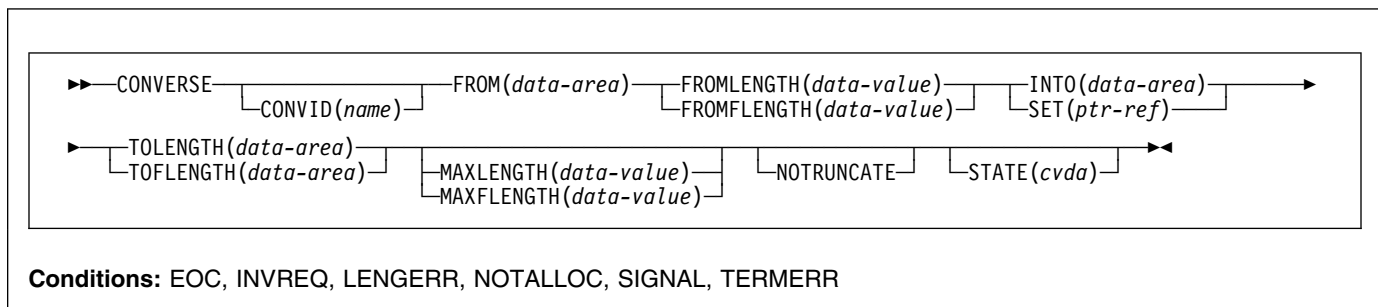


Description

This form of the CONVERSE command is used by all CICS-supported VTAM terminals for which the other CONVERSE descriptions are not appropriate.

CONVERSE (APPC)

Communicate on an APPC mapped conversation.

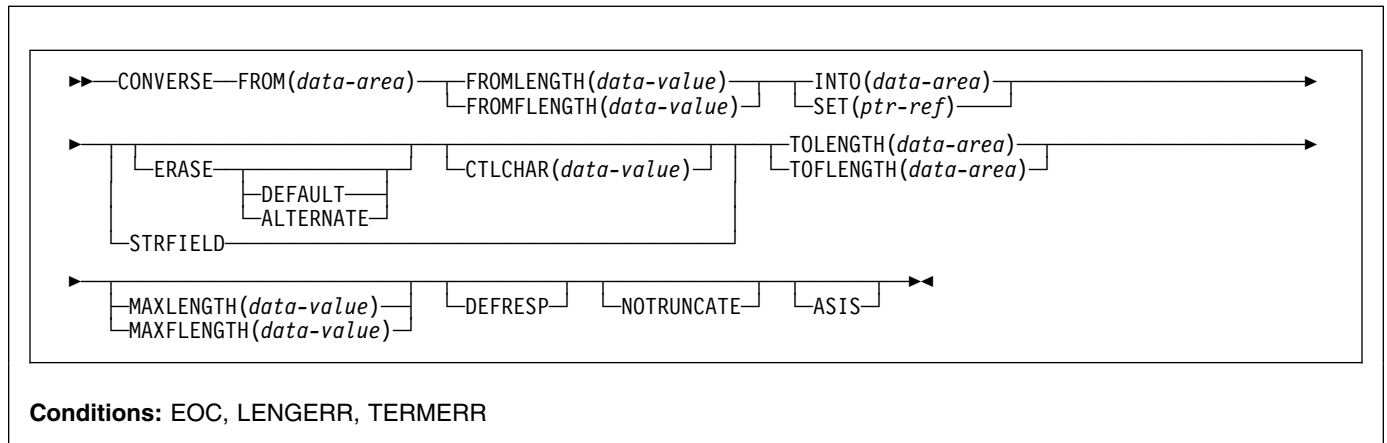


Description

CONVERSE sends, then receives, data on an APPC mapped conversation.

CONVERSE (LUTYPE2/LUTYPE3)

Communicate on a 3270-display logical unit (LUTYPE2) or 3270-printer logical unit (LUTYPE3).

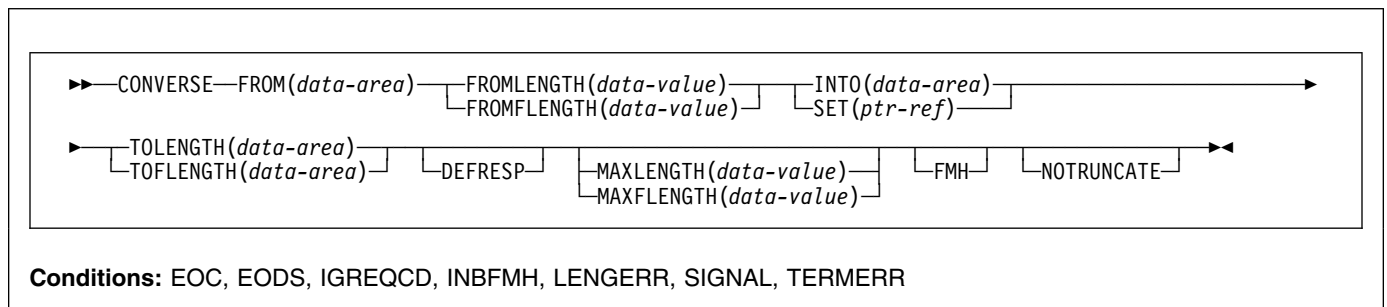


Description

CONVERSE communicates on a 3270-display logical or 3270-printer logical unit.

CONVERSE (LUTYPE4)

Communicate on an LUTYPE4 logical unit.



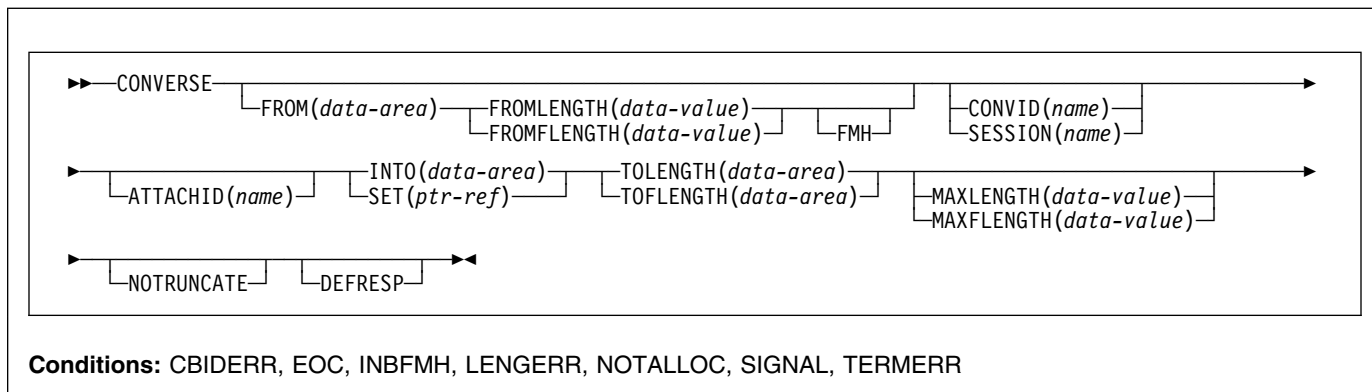
Description

CONVERSE communicates on an LUTYPE4 logical unit.

CONVERSE (VTAM)

CONVERSE (LUTYPE6.1)

Communicate on an LUTYPE6.1 logical unit.

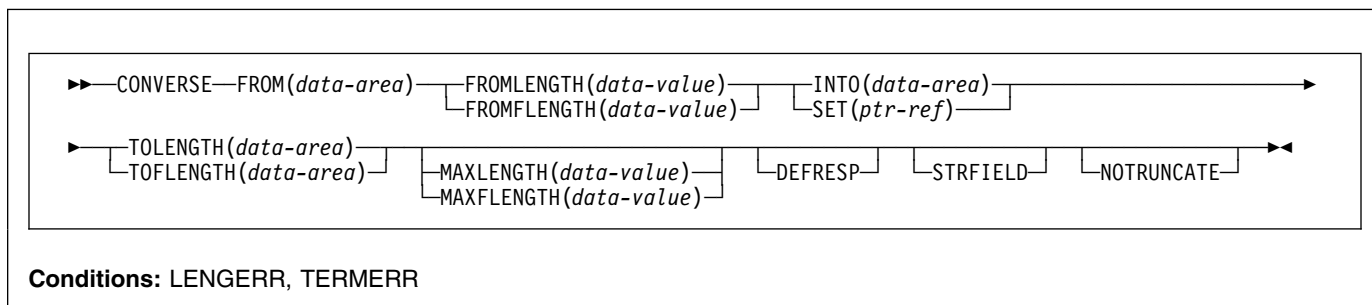


Description

CONVERSE communicates on an LUTYPE6.1 logical unit.

CONVERSE (SCS)

Communicate on a 3270 SCS printer logical unit.

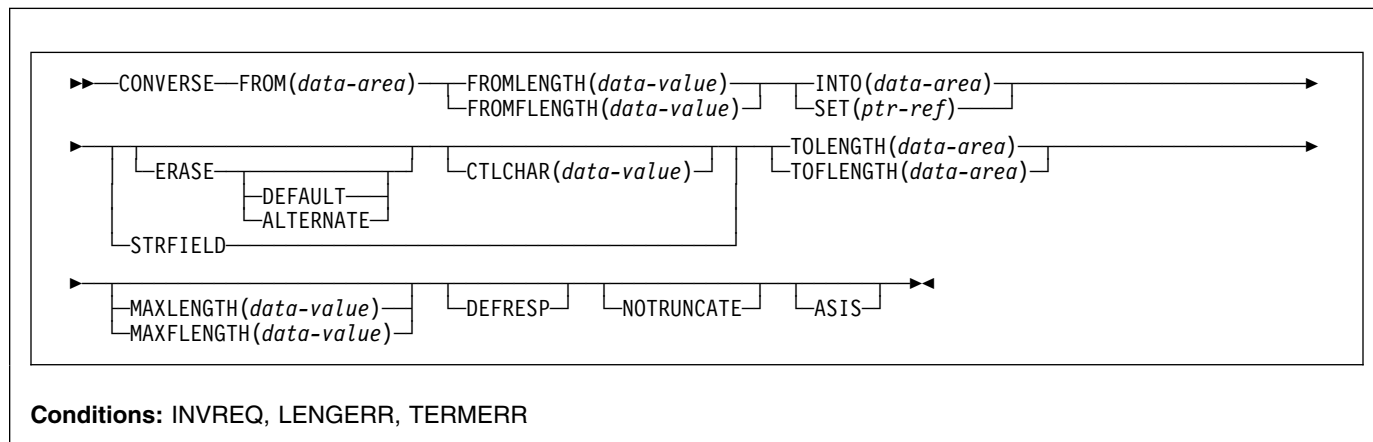


Description

CONVERSE communicates on a 3270 SNA character string (SCS) printer logical unit. The SCS printer logical unit accepts a character string as defined by Systems Network Architecture (SNA). Some devices connected under SNA can send a signal that can be detected by the HANDLE CONDITION SIGNAL command, which in turn can invoke an appropriate handling routine. If necessary, a WAIT SIGNAL command can be used to make the application program wait for the signal. The PA keys on a 3287 can be used in this way, or with a RECEIVE command.

CONVERSE (3270 logical)

Communicate on a 3270 logical unit.

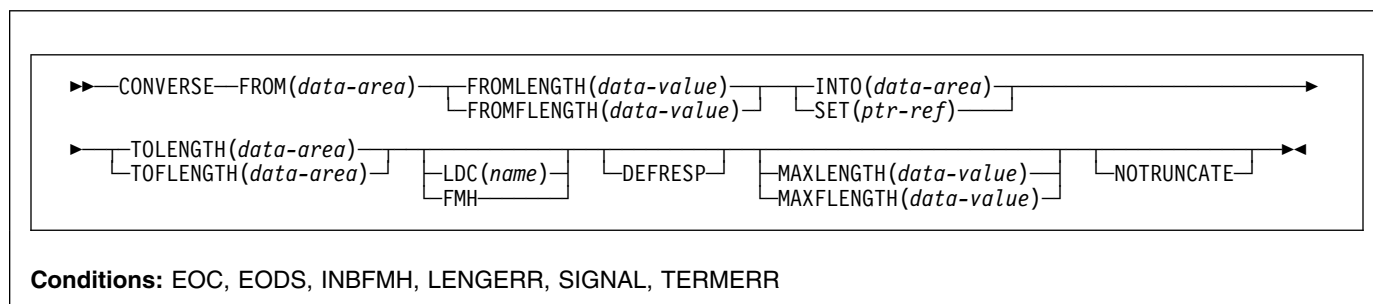


Description

CONVERSE communicates on a 3270 logical unit.

CONVERSE (3600-3601)

Communicate on a 3600 (3601) logical unit.



Description

CONVERSE communicates on a 3600 logical unit. This form of the CONVERSE command also applies to the 3630 plant communication system.

A logical device code (LDC) is a code that can be included in an outbound Function Management Header (FMH) to specify the disposition of the data (for example, to which subsystem terminal it should be sent). Each code can be represented by a unique LDC mnemonic.

The installation can specify up to 256 2-character mnemonics for each TCTTE, and two or more TCTTEs can share a list of these mnemonics. A numeric value (0 through 255) corresponds to each LDC mnemonic for each TCTTE.

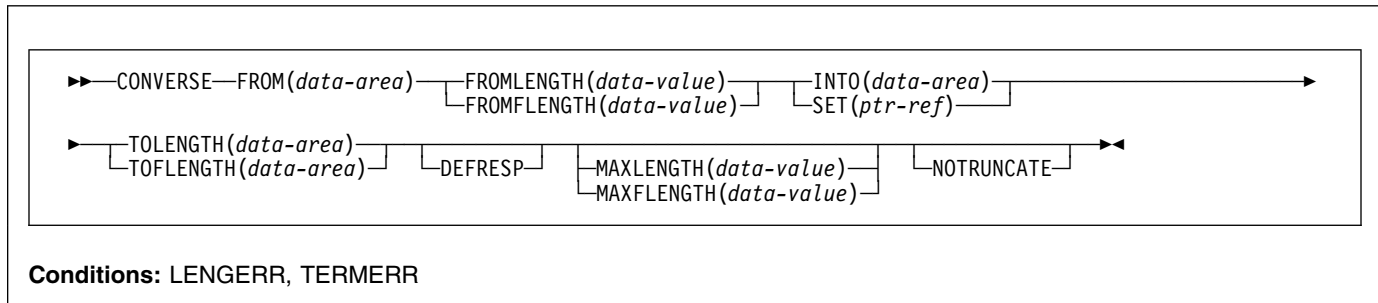
CONVERSE (VTAM)

A 3600 device and a logical page size are also associated with an LDC. “LDC” or “LDC value” is used in this book to refer to the code specified by the user; “LDC mnemonic” refers to the 2-character symbol that represents the LDC numeric value.

When the LDC option is specified in the CONVERSE command, the numeric value associated with the mnemonic for the particular TCTTE is inserted in the FMH. This value is chosen by the installation, and is interpreted by the 3601 application program.

CONVERSE (3600-3614)

Communicate on a 3600 (3614) logical unit.



Description

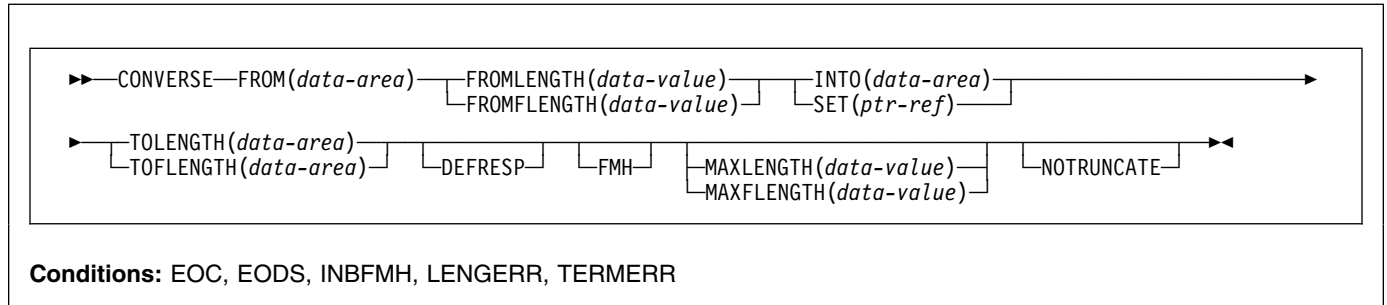
CONVERSE communicates on a 3600 logical unit.

The data stream and communication format used between a CICS application program and a 3614 is determined by the 3614. The application program is therefore device_dependent when handling 3614 communication.

For further information about designing 3614 application programs for CICS, refer to the *CICS/DOS/VS IBM 4700/3600/3630 Guide*.

CONVERSE (3650 interpreter)

Communicate on a 3650 interpreter logical unit.

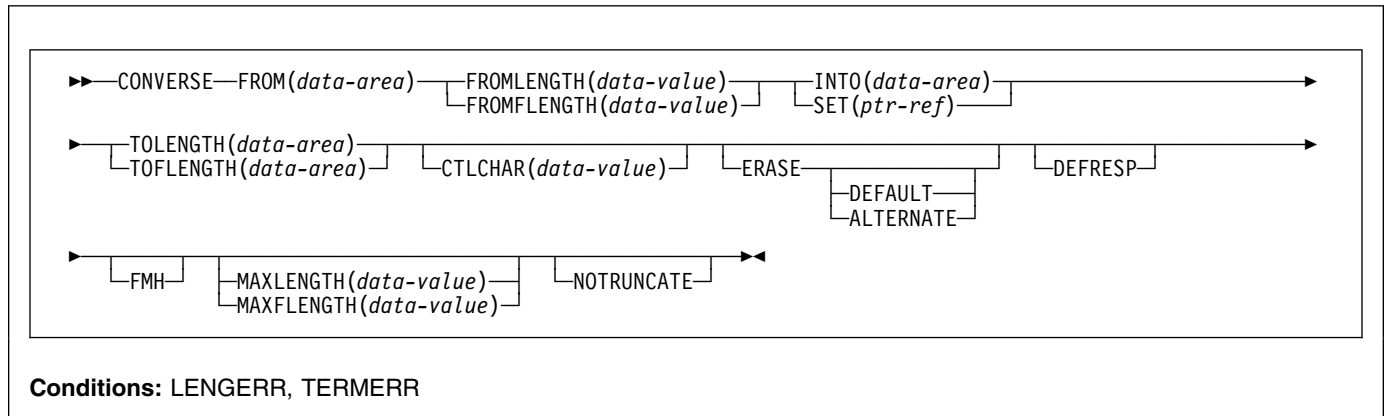


Description

CONVERSE communicates on a 3650 interpreter logical unit.

CONVERSE (3650-3270)

Communicate on a 3650 host conversational (3270) logical unit.



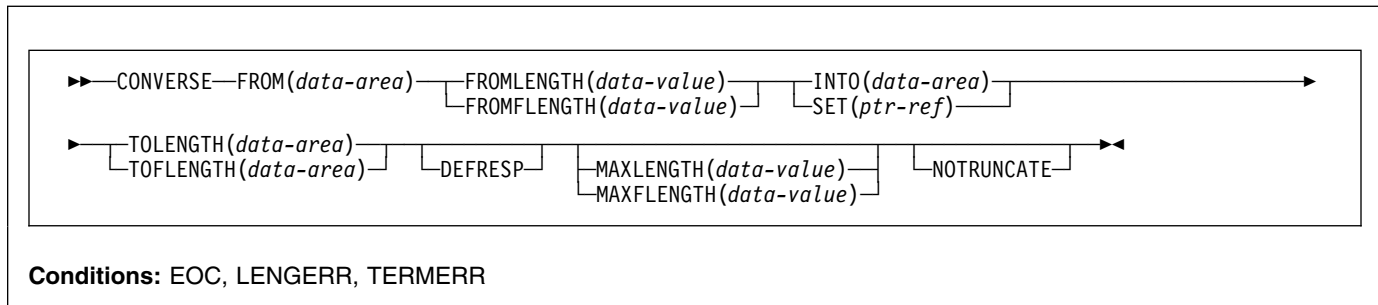
Description

CONVERSE communicates on a 3650 host conversational logical unit.

CONVERSE (VTAM)

CONVERSE (3650-3653)

Communicate on a 3650 host conversational (3653) logical unit.

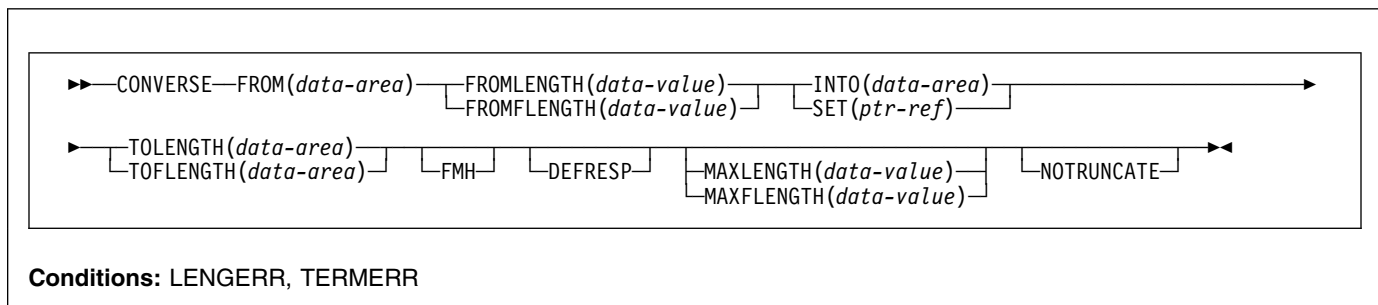


Description

CONVERSE communicates on a 3650 host conversational logical unit.

CONVERSE (3650-3680)

Communicate on a 3650 host command processor (3680) logical unit.

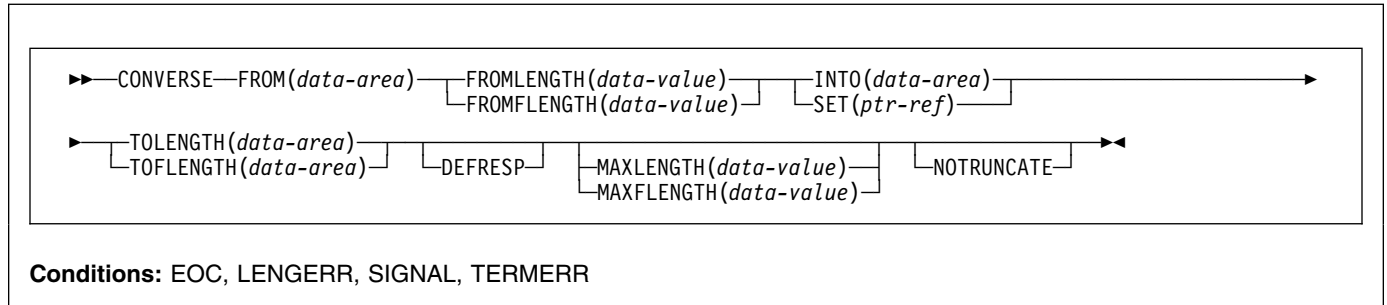


Description

CONVERSE communicates on a 3650 host command processor logical unit.

CONVERSE (3767)

Communicate on a 3767 interactive logical unit.

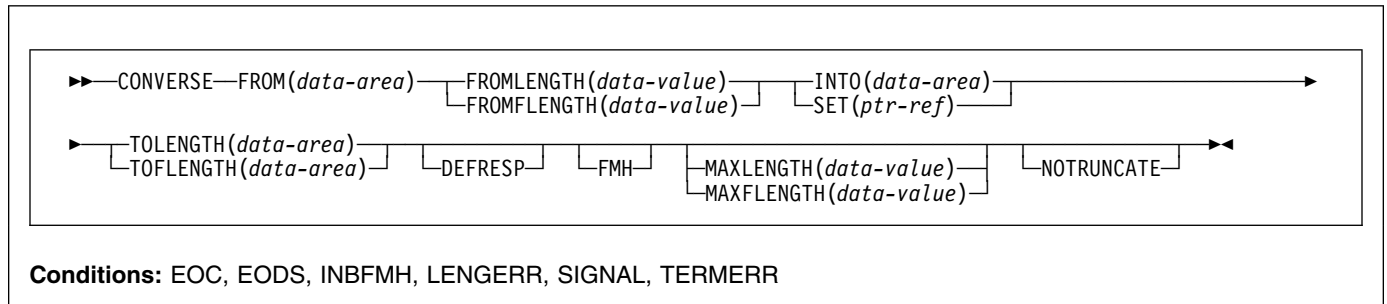


Description

CONVERSE communicates on a 3767 interactive logical unit. This command also applies to the 3770 interactive logical unit.

CONVERSE (3770)

Communicate on a 3770 batch logical unit.



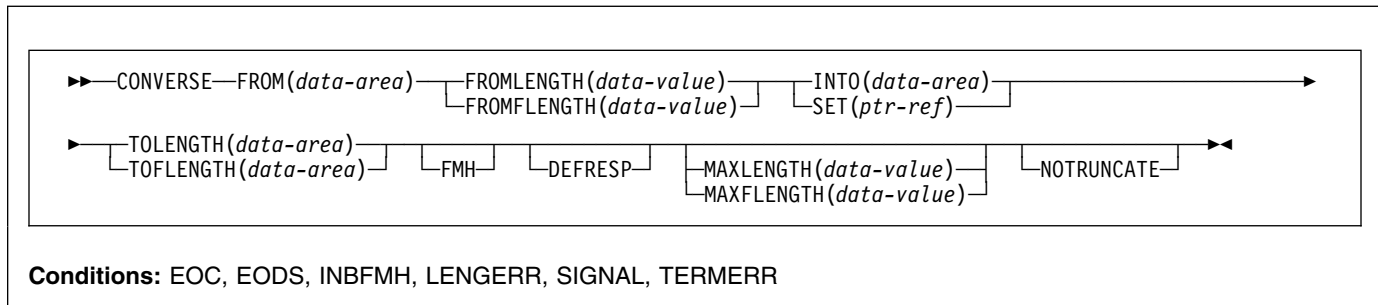
Description

CONVERSE communicates on a 3770 batch logical unit.

CONVERSE (VTAM)

CONVERSE (3790 full-function or inquiry)

Communicate on a 3790 full-function or inquiry logical unit.

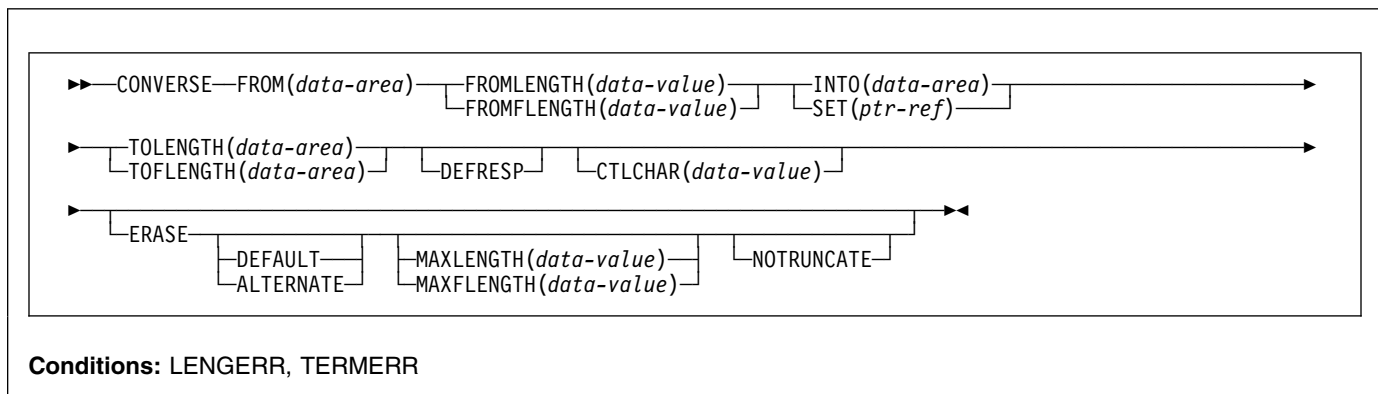


Description

CONVERSE communicates on a 3790 full-function or inquiry logical unit.

CONVERSE (3790 3270-display)

Communicate on a 3790 (3270-display) logical unit.



Description

CONVERSE communicates on a 3790 logical unit.

Options

Note: CICS Transaction Server for VSE/ESA, unlike CICS/ESA®, does not support the DEST option. If a program with this option is translated on VSE, a warning message DFH7275 is produced and the option is ignored.

ALTERNATE

sets the terminal to use the ALTERNATE screen size.

ASIS

indicates that output is to be sent in transparent mode (with no recognition of control characters and accepting any of the 256 possible combinations of eight bits as valid transmittable data).

Note: If you are using a katakana terminal, you might see some messages containing mixed English and katakana characters. That is because katakana terminals cannot display mixed-case output. Uppercase characters in the data stream appear as uppercase English characters, but lowercase characters appear as katakana characters. If this happens, ask your system programmer to specify MSGCASE=UPPER in the system initialization parameters so that messages contain uppercase characters only. This note applies to any command that is used to receive katakana characters, not just to CONVERSE commands.

ATTACHID(*name*)

specifies that an attach header (created by a BUILD ATTACH command) is to precede, and be concatenated with, the user data supplied in the FROM option. “name” (1–8 characters) identifies the attach header control block to be used in the local task.

CONVID(*name*)

identifies the conversation to which the command relates. The 4-character name identifies either the token returned by a previously executed ALLOCATE command in EIBRSRCE in the EIB, or the token representing the principal session (returned by a previously executed ASSIGN command).

For compatibility with earlier releases, SESSION is accepted as a synonym for CONVID. New programs should use CONVID.

If the option is omitted, the principal facility for the task is used by default.

CTLCHAR(*data-value*)

specifies a 1-byte write control character (WCC) that controls the CONVERSE command. A COBOL user must specify a data area containing this character.

If the option is omitted, all modified data tags are reset to zero, and the keyboard is restored.

DEFAULT

sets the terminal to use the DEFAULT screen size.

DEFRESP

indicates that a definite response is required when the output operation has been completed.

ERASE

specifies that the screen printer buffer or partition is to be erased and the cursor returned to the upper left corner of the screen. (This option applies only to the 3270, or 8775, and to the 3604 Keyboard Display.)

The first output operation in any transaction, or in a series of pseudoconversational transactions, should always specify ERASE. For transactions attached to 3270 screens or printers, unless explicitly overridden by the DEFAULT or ALTERNATE option, this also ensures that the correct screen size is selected, as defined for the transaction by the SCRNSIZE option in the RDO PROFILE definition.

FMH

specifies that a function management header has been included in the data to be written. If the ATTACHID option is specified as well, the concatenated FMH flag is set in the attach FMH.

FROM(*data-area*)

specifies the data to be written to the terminal or logical unit, or sent to the partner transaction. This option may, when relevant, be omitted if ATTACHID is specified.

FROMLENGTH(*data-value*)

is a fullword alternative to FROMLENGTH.

FROMLENGTH(*data-value*)

specifies the length, as a halfword binary value, of the data. For a description of a safe upper limit, see “LENGTH options” on page 5.

INTO(*data-area*)

specifies the receiving field for the data read from the terminal or logical unit, or the application target data area into which data is to be received from the application program connected to the other end of the current conversation.

LDC(*name*)

specifies the 2-character mnemonic used to determine the appropriate logical device code (LDC) numeric value. The mnemonic identifies an LDC entry defined by a DFHTCT TYPE=LDC macro. See the *CICS Resource Definition Guide* for further information about the DFHTCT macro.

MAXLENGTH(*data-value*)

is a fullword alternative to MAXLENGTH.

MAXLENGTH(*data-value*)

specifies the maximum amount (halfword binary value) of data that CICS is to recover in response to a CONVERSE (default) command. If INTO is specified, MAXLENGTH overrides the use of TOLENGTH as an input to CICS. If SET is specified, MAXLENGTH provides a way for the program to limit the amount of data it receives at one time.

If the value specified is less than zero, zero is assumed.

CONVERSE (VTAM)

If the length of data exceeds the value specified and the NOTRUNCATE option is not present, the data is truncated to that value and the LENGERR condition occurs. The data area specified in the TOLENGTH option is set to the original length of data.

If the length of data exceeds the value specified and the NOTRUNCATE option is present, CICS retains the remaining data and uses it to satisfy subsequent RECEIVE commands. The data area specified in the TOLENGTH option is set to the length of data returned.

If no argument is coded for MAXLENGTH, CICS defaults to TOLENGTH.

NOTRUNCATE

specifies that, when the data available exceeds the length requested, the remaining data is not to be discarded but is to be retained for retrieval by subsequent RECEIVE commands.

SESSION(*name*)

specifies the symbolic identifier (1–4 characters) of a session TCTTE. This option specifies the alternate facility to be used. If both this option and CONVID are omitted, the principal facility for the task is used.

SET(*ptr-ref*)

specifies the pointer reference to be set to the address of the data read from the terminal. pointer reference, unless changed by other commands or statements, is valid until the next CONVERSE (default) command or the end of task.

If DATALOCATION(ANY) is associated with the application program, the address of the data can be above or below the 16MB line.

If DATALOCATION(BELOW) is associated with the application program, and the data resides above the 16MB line, the data is copied below the 16MB line, and the address of this copy is returned.

If TASKDATAKEY(USER) is specified for the running task, and storage protection is active, the data returned is in a user-key. If TASKDATAKEY(CICS) is specified and storage protection is active, the data returned is in a CICS-key.

STATE(*cvda*)

gets the state of the current conversation. The *cvda* values returned by CICS are:

ALLOCATED
CONFFREE
CONFRECEIVE
CONFSEND
FREE
PENDFREE
PENDRECEIVE
RECEIVE
ROLLBACK
SEND
SYNCFREE
SYNCRECEIVE

SYNCSND

STRFIELD

specifies that the data area specified in the FROM option contains structured fields. If this option is specified, the contents of all structured fields must be handled by the application program. The CONVERSE command must be used if the data area contains a read partition structured field. (Structured fields are described in the *CICS 3270 Data Stream Device Guide*.)

CTLCHAR and ERASE are mutually exclusive with STRFIELD, and their use with STRFIELD generates an error message.

TOLENGTH(*data-area*)

is a fullword alternative to TOLENGTH.

TOLENGTH(*data-area*)

specifies the length (halfword binary value) of the data to be received. If you specify INTO, but omit MAXLENGTH, “*data-area*” specifies the maximum length that the program accepts. If the value is less than zero, zero is assumed.

If the length of the data exceeds the value specified, but NOTRUNCATE is omitted, the data is truncated to that value, and the LENGERR condition occurs. When the data is received, the data area is set to the length of the data.

For a description of a safe upper limit, see “LENGTH options” on page 5.

Conditions

Some of the following conditions can occur in combination with others. CICS checks for these conditions in the following order:

1. EODS
2. INBFMH
3. EOC.

If more than one occurs, only the first is passed to the application program. EIBRCODE, however, is set to indicate all the conditions that occurred.

CBIDERR

occurs if the requested attach header control block named in ATTACHID cannot be found.

Default action: terminate the task abnormally.

EOC

occurs when a request/response unit (RU) is received with the end-of-chain indicator set. Field EIBEEOC also contains this indicator.

Default action: ignore the condition.

EODS

occurs when an end-of-data-set indicator is received.

Default action: terminate the task abnormally.

IGREQCD

occurs when an attempt is made to execute a CONVERSE command after a SIGNAL data-flow control command with a request change direction (RCD) code has been received from an LUTYPE4 logical unit.

Default action: terminate the task abnormally.

INBFMH

occurs if a request/response unit (RU) contains a function management header (FMH). Field EIBFMH contains this indicator and it should be used in preference to INBFMH. The IGNORE CONDITION command can be used to ignore the condition.

Default action: terminate the task abnormally.

INVREQ

RESP2 values:

200 A distributed program link server application specified the function shipping session (its principal facility)

also occurs (RESP2 not set) in any of the following situations:

- The command is used on a conversation that is in use by CPI Communications, or that is an APPC basic conversation. In the latter case, the application should have issued a GDS SEND INVITE followed by a GDS RECEIVE.

Default action: terminate the task abnormally.

LENGERR

occurs in any of the following situations:

- Data received is discarded by CICS because its length exceeds the maximum that the program accepts (see TOLENGTH and MAXLENGTH options), and the NOTTRUNCATE option is not specified.
- An out-of-range value is supplied in one of the options, FROMLENGTH, FROMFLENGTH, MAXLENGTH, MAXFLENGTH, TOLENGTH, or TOFLENGTH.

Default action: terminate the task abnormally.

NOTALLOC

occurs if the facility specified in the command is not owned by the application, or does not relate to a conversation owned by the application.

Default action: terminate the task abnormally.

SIGNAL

occurs when an inbound SIGNAL data-flow control command is received from a logical unit or session, or the partner transaction. EIBSIG is always set when an inbound signal is received.

Default action: ignore the condition.

TERMERR

occurs for a terminal or session-related error. Any action on that conversation other than a FREE causes an ATCV abend.

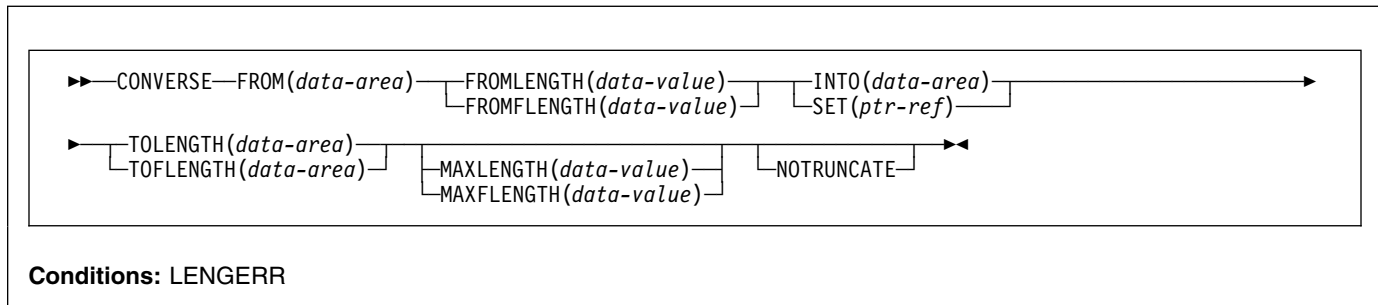
A CANCEL TASK request by a user node error program (NEP) may cause this condition if the task has an outstanding terminal control request active when the node abnormal condition program handles the session error.

Default action: terminate the task abnormally with abend code ATNI.

CONVERSE (non-VTAM)

CONVERSE (non-VTAM default)

Communicate on standard CICS terminal support.

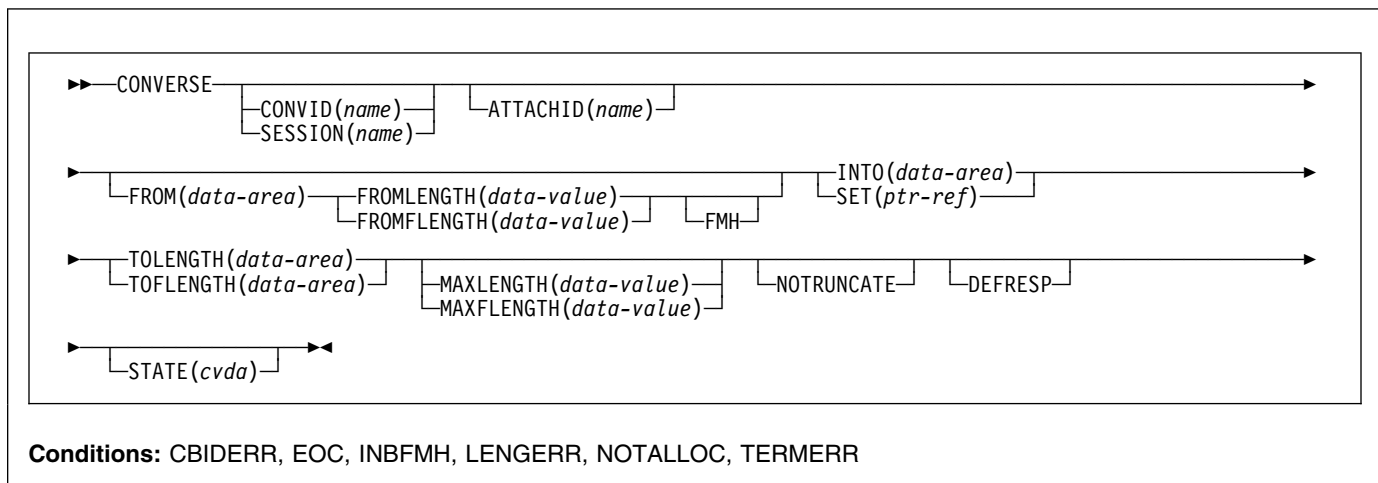


Description

This form of the CONVERSE command is used by all CICS-supported terminals for which the other CONVERSE descriptions are not appropriate.

CONVERSE (MRO)

Communicate on an MRO session.

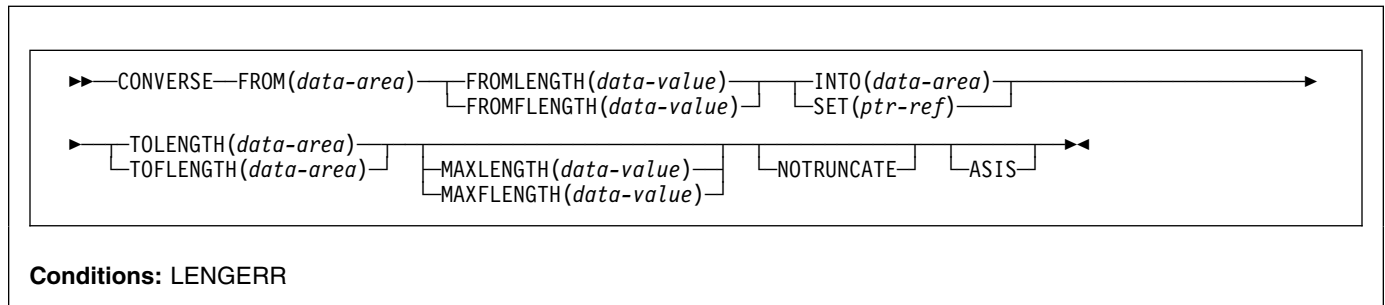


Description

CONVERSE communicates on an MRO session. For more information about MRO and IRC, see the *CICS Intercommunication Guide*.

CONVERSE (System/3)

Communicate on a System/3 terminal.



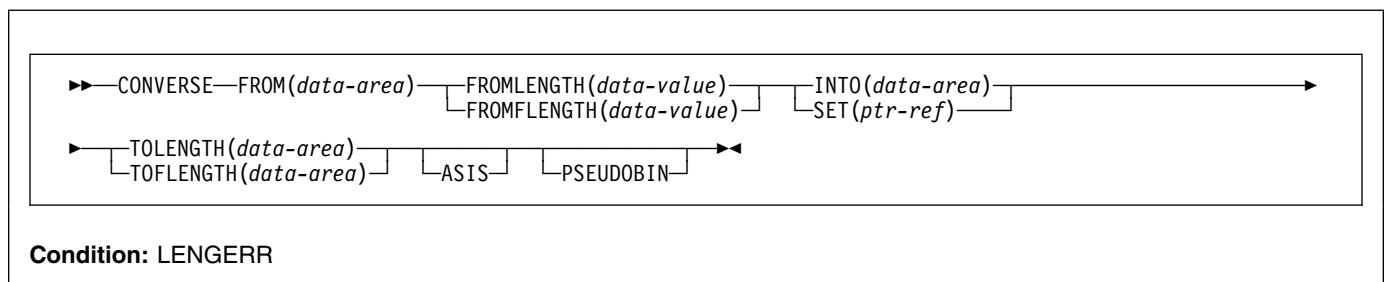
Description

CONVERSE communicates on a System/3 terminal. CONVERSE in this form also applies to the following devices:

- 2770 data communication system
- 2780 data transmission terminal
- 3660 supermarket scanning system
- 3780 communication terminal.

CONVERSE (System/7)

Communicate on a System/7 terminal.



Description

CONVERSE communicates on a System/7 terminal.

Transactions are normally initiated from the System/7 by issuing a 4-character transaction code that is transmitted in BCD mode. Pseudobinary mode can be used only while communicating with an active CICS transaction; it cannot be used to initiate the transaction. The message length is given as the number of words to be transmitted (not as the number of characters).

When a transaction is initiated on a System/7, CICS services that System/7 only for the duration of the transaction; that is, to ensure efficient use of the line, any other System/7s on the same line are locked out for the duration of the transaction. CICS application programs for the multipoint System/7 should be designed with the shortest possible execution time.

CONVERSE (non-VTAM)

The first word (two characters) of every message sent to the System/7 must be an identification word, except words beginning with “@”, which are reserved by CICS.

When the PSEUDOBIN option is specified on RECEIVE or SEND, the length of the data area provided by the application program must be at least twice that of the data to be read.

In the case of a System/7 on a dial-up (switched) line, the System/7 application program must, initially, transmit a four-character terminal identification. (This terminal identification is generated during preparation of the TCT through use of the DFHTCT TYPE=TERMINAL, TRMIDNT=parameter specification.) CICS responds with either a ‘ready’ message, indicating that the terminal identifier is valid and that the System/7 may proceed as if it were on a leased line, or an ‘INVALID TERMINAL IDENTIFICATION’ message, indicating that the terminal identifier sent by the System/7 did not match the TRMIDNT= parameter specification.

Whenever CICS initiates the connection to a dial-up System/7, CICS writes a null message, consisting of three idle characters, prior to starting the transaction. If there is no program resident in the System/7 capable of supporting the Asynchronous Communication Control Adapter (ACCA), BTAM error routines cause a data check message to be recorded on the CICS (host) system console. This is normal if the task initiated by CICS is to IPL the System/7. Although the data check message is printed, CICS ignores the error and continues normal processing. If a program capable of supporting the ACCA is resident in the System/7 at the time this message is transmitted, no data check occurs.

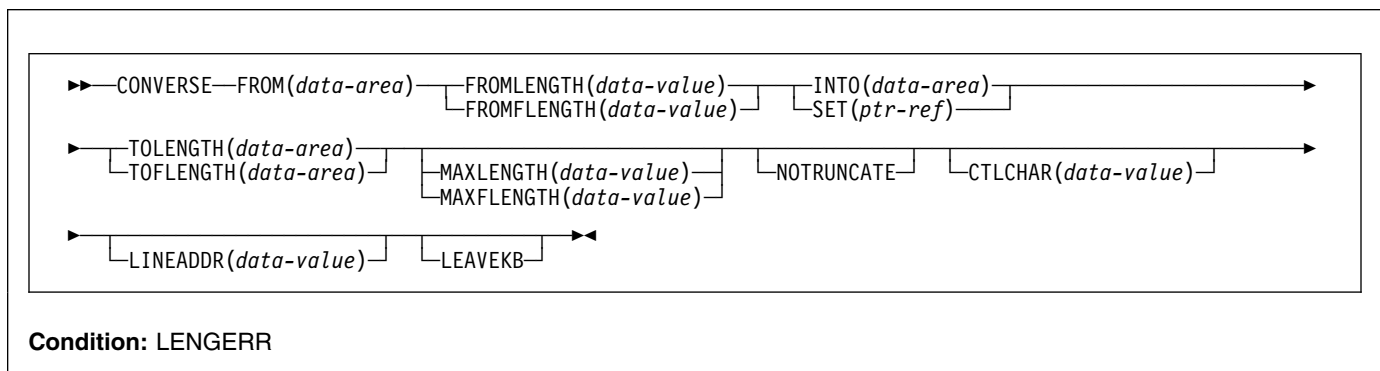
When a disconnect is issued to a dial-up System/7, the “busy” bit is sometimes left on in the interrupt status word of the ACCA. If the line connection is reestablished by dialing from the System/7 end, the ‘busy’ condition of the ACCA prevents message transmission from the System/7. To overcome this problem, the System/7 program must reset the ACCA after each disconnect and before message transmission is attempted. This can be done by issuing the following instruction:

```
PWRI 0,8,3,0 RESET ACCA
```

This procedure is not necessary when the line is reconnected by CICS (that is, by an automatically initiated transaction).

CONVERSE (2260)

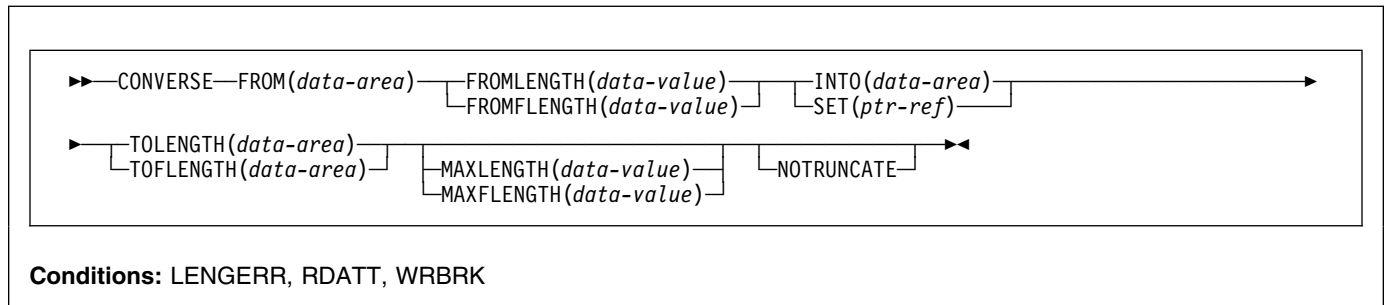
Communicate on a 2260 or 2265 display station.



CONVERSE communicates on a 2260 or 2265 display station.

CONVERSE (2741)

Communicate on a 2741 communication terminal.



Description

CONVERSE communicates on a 2741 terminal.

Read attention

If the terminal operator presses the attention key on the 2741 after typing a message, it is recognized as a read attention if:

- Read attention is supported by the system
- The message is read by a RECEIVE command.

When this occurs, control is transferred to a CICS read-attention exit routine, if it has been generated into the system. This routine is a skeleton program that can be tailored by the system programmer to carry out actions such as the following:

- Perform data analysis or modification on a read attention.
- Return a common response to the terminal operator following a read attention.
- Return a response and request additional input that can be read into the initial input area or into a new area.
- Request new I/O without requiring a return to the task to request additional input.

When the read-attention exit routine is completed, control is returned to the application program at the address specified in the HANDLE CONDITION RDATT command. The return is made when either of the following occurs:

- The exit routine issues no more requests for input.
- The exit routine issues a RECEIVE request and the operator terminates the input with a carriage return. (If the operator terminates the input with an attention, the exit routine is reentered and is free to issue another RECEIVE request.)

If a HANDLE CONDITION RDATT command is not included in the application program or read attention is not supported, the attention is treated as if the return key had been pressed.

Write break

If the terminal operator presses the attention key on the 2741 while a message is being received, it is recognized as a write break if:

- Write break is supported by the system.
- A HANDLE CONDITION WRBRK command is active in the application program.

When this occurs, the remaining portion of the message is not sent to the terminal. The write is terminated as though it were successful, and a new-line character (X'15') is sent to cause a carrier return. Control is returned to the application program at the address specified for the WRBRK condition.

CONVERSE (non-VTAM)

If a HANDLE CONDITION WRBRK command is not included in the application program or if write break is not supported, the break is treated as an I/O error.

CONVERSE (2770)

Description

The support and command syntax for 2770 is the same as for System/3, see “CONVERSE (System/3)” on page 53.

The 2770 recognizes a read interrupt and responds by transmitting the contents of the I/O buffer. After the contents of the buffer have been transmitted, the 2770 responds to the next read continue with an EOT. If the I/O buffer is empty, the 2770 transmits an EOT. CICS issues a read interrupt and read continue to relinquish use of the line and to enable the application program to write to the 2770.

Input from a 2770 consists of one or more logical records. CICS provides one logical record for each read request to the application program. The size of a logical record cannot exceed the size of the I/O buffer. If the input spans multiple buffers, multiple reads must be issued by the application program.

The 2265 component of the 2770 Data Communication System is controlled by data stream characters, not by BTAM macros; appropriate screen control characters should be included in the output area.

For 2770 input, data is deblocked to ETX, ETB, RS, and US characters. These characters are moved with the data to the input area but are not included in the data length; characters such as NL, CR, and LF are passed in the input area as data.

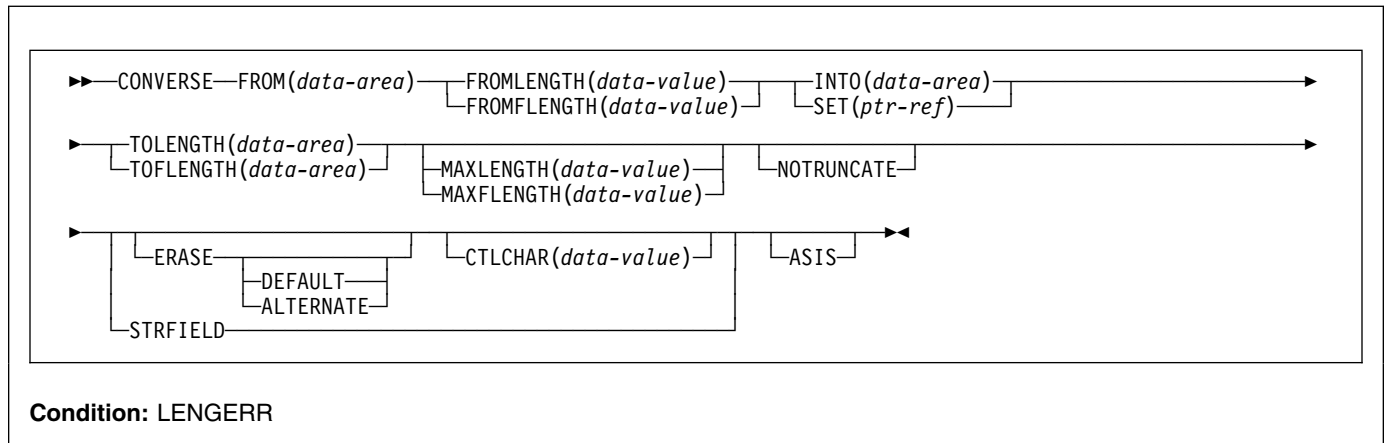
CONVERSE (2780)

The support and command syntax for 2780 is the same as for System/3, see “CONVERSE (System/3)” on page 53.

Output to a 2780 requires that the application program contains an appropriate “escape sequence” for component selection associated with the output message. For details of the 2780, see the publication *IBM 2780 Data Transmission Terminal: Component Description* manual.

CONVERSE (3270 display)

Communicate on a 3270 information display system (BTAM) - valid only for remote terminals.

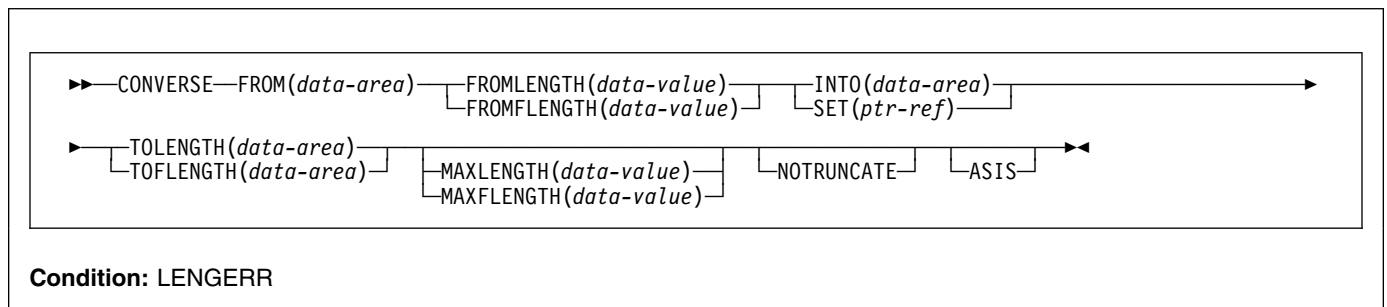


Description

CONVERSE communicates on a 3270 information display system.

CONVERSE (3600 BTAM)

Communicate on a 3600 finance communication system (BTAM) - only valid for remote terminals.



Description

CONVERSE communicates on a 3600 finance communication system. This form of the CONVERSE command also applies to the 4700 finance communication system. See "BTAM programmable terminals" on page 388.

CONVERSE (non-VTAM)

Input

The unit of transmission from a 3601 Finance Communication Controller to CICS is a segment consisting of the start-of-text data-link control character (STX), the 1-byte identification of the 3600 logical workstation that issued the processor write, the data, and either an end-of-block (ETB) or an end-of-text (ETX) control character.

A logical workstation sends a message either in one segment, in which case the segment ends with ETX, or in more than one segment, in which case only the last segment ends with ETX, all others ending with ETB.

The input area passed to the user-written application program consists of the data only. The 1-byte field TCTTEDLM, which can be obtained using an ASSIGN DELIMITER command, contains flags describing the data-link control character (ETB, ETX, or IRS) that ended the segment. The application program can issue terminal control commands to read the data until it receives a segment ending with ETX. If blocked data is transmitted, it is received by CICS as blocks of segments. Only the first segment in a block starts with the STX control character, and all segments are separated by IRS characters. None of the segments contains ETB or ETX characters except the last, which has the ETX character.

For blocked input, the flags in TCTTEDLM only indicate end of segment, not end of message. The CICS application program still receives only the data, but user-defined conventions may be required to determine the end of the message.

The field TCTTEDLM also indicates the mode of the input, either transparent or nontransparent. Blocked input is nontransparent.

The terminal control facility does not pass input containing a start-of-header (SOH) data-link control character to a user-written application program. If it receives an SOH, it sets an indicator in TCTTEDLM, passes the input to the user exit in the terminal control facility, and then discards it.

Output

When an application program issues a SEND command, the terminal control facility determines, from the value specified in the BUFFER parameter of the DFHTCT TYPE=TERMINAL system macro, the number of segments to be built for the message. It sends the message to the 3600 logical unit either in one segment consisting of a start-of-text character (STX), the data, and an end-of-text character (ETX); or in more than one segment, in which case only the last ends with ETX, all others ending with ETB.

The host input buffer of the 3600 controller and the input segment of the receiving logical unit must be large enough to accommodate the data sent by CICS. However, space for the data-link control characters need not be included. The 3600 application program reads the data from the host, by means of an LREAD, until it has received the entire message.

CICS system output messages begin with DFH followed by a 2-byte component id, 4-byte message number and the message text. These messages are sent in nontransparent mode. CICS user-written application programs should not send messages starting with DFH to the 3601.

Resend message

When a logical unit sends a message to the host and a 'short on storage' condition exists or the input is unsolicited (the active task associated with the terminal has not issued a read), the terminal control facility sends a "resend" message to the logical unit. No message is sent to the destinations CSMT or CSTL.

The first eight bytes of data sent to CICS can be used by the 3600 application program to define a convention to associate responses received from CICS with transactions sent to the host; for example, sequence numbers could be used.

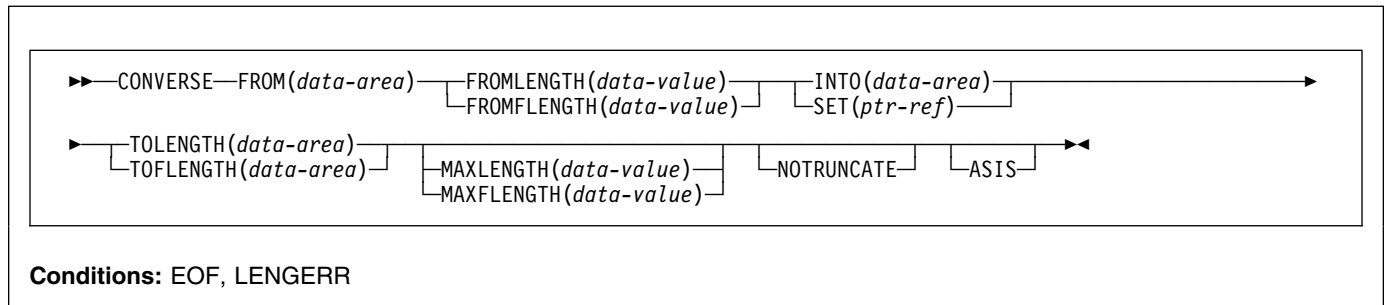
If a CICS user-written application program has already issued a SEND command when a resend situation occurs, the resend message is not sent to the 3601 until the user-written application program message has been sent. A 3600 logical unit cannot receive a resend message while receiving a segmented message.

Only one resend message at a time can be queued for a logical unit. If a second resend situation occurs before CICS has written the first, a resend message is sent containing the eight bytes of data that accompanied the second input transaction from the 3600 logical unit.

The resend message is sent in transparent mode if the input data from the 3601 to be retransmitted is received by CICS in transparent mode. Otherwise it is sent in nontransparent mode.

CONVERSE (3735)

Communicate on a 3735 programmable buffered terminal.

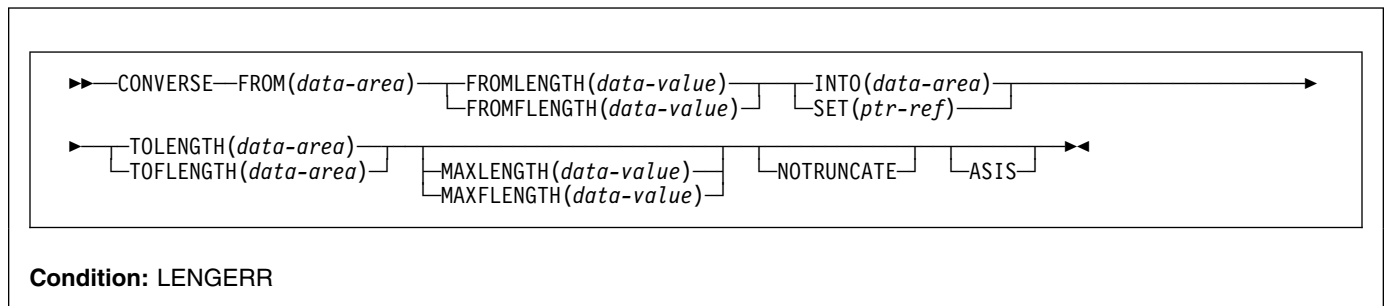


Description

CONVERSE communicates on a 3735 programmable buffered terminal. The 3735 programmable buffered terminal may be serviced by CICS in response to terminal-initiated input (automatic answering), or as a result of an automatic calling or time-initiated transaction.

CONVERSE (3740)

Communicate on a 3740 data entry system.



Description

CONVERSE communicates on a 3740 data entry system.

In batch mode, many files are exchanged between the 3740 and CICS in a single transmission. The transmission of an input batch must be complete before an output transmission can be started.

CONVERSE (non-VTAM)

Options

Note: CICS Transaction Server for VSE/ESA, unlike CICS/ESA, does not support the DEST option. If a program with this option is translated on VSE, a warning message DFH7275 is produced and the option is ignored.

ALTERNATE

set the terminal to use the ALTERNATE screen size.

ASIS

indicates that output is to be sent in transparent mode (with no recognition of control characters and accepting any of the 256 possible combinations of eight bits as valid transmittable data).

Note: If you are using a katakana terminal, you might see some messages containing mixed English and katakana characters. That is because katakana terminals cannot display mixed-case output. Uppercase characters in the data stream appear as uppercase English characters, but lowercase characters appear as katakana characters. If this happens, ask your system programmer to specify MSGCASE=UPPER in the system initialization parameters so that messages contain uppercase characters only.

ATTACHID(*name*)

specifies that an attach header (created by a BUILD ATTACH command) is to precede, and be concatenated with, the user data supplied in the FROM option. "name" (1–8 characters) identifies the attach header control block to be used in the local task.

CONVID(*name*)

identifies the conversation to which the command relates. The 4-character name identifies either the token returned by a previously executed ALLOCATE command in EIBRSRCE in the EIB, or the token representing the principal session (returned by a previously executed ASSIGN command).

CTLCHAR(*data-value*)

specifies a 1-byte write control character (WCC¹) that controls the CONVERSE command. A COBOL user must specify a data area containing this character. If the option is omitted, all modified data tags are reset to zero and the keyboard is restored.

DEFAULT

set the terminal to use the DEFAULT screen size.

DEFRESP

indicates that a definite response is required when the output operation has been completed.

ERASE

specifies that the screen printer buffer or partition is to be erased and the cursor returned to the upper left

corner of the screen. (This option applies only to the 3270, or 8775, and to the 3604 Keyboard Display.)

The first output operation in any transaction, or in a series of pseudoconversational transactions, should always specify ERASE. For transactions attached to 3270 screens or printers, unless explicitly overridden by the DEFAULT or ALTERNATE option, this also ensures that the correct screen size is selected, as defined for the transaction by the SCRNSIZE option in the RDO PROFILE definition.

FMH

specifies that a function management header has been included in the data to be written. If the ATTACHID option is specified as well, the concatenated FMH flag is set in the attach FMH.

FROM(*data-area*)

specifies the data to be written to the terminal or logical unit, or sent to the partner transaction. This option may, when relevant, be omitted if ATTACHID is specified.

FROMLENGTH(*data-value*)

is a fullword alternative to FROMLENGTH.

FROMLENGTH(*data-value*)

specifies the length, as a halfword binary value, of the data to be written. If you use this option, you must also specify FROM. For a description of a safe upper limit, see "LENGTH options" on page 5.

INTO(*data-area*)

specifies the receiving field for the data read from the logical unit or terminal.

LEAVEKB

specifies that the keyboard is to remain locked at the completion of the data transfer.

LINEADDR(*data-value*)

specifies that the writing is to begin on a specific line of a 2260/2265 screen. The data value is a halfword binary value in the range 1 through 12 for a 2260, or 1 through 15 for a 2265.

MAXLENGTH(*data-value*)

is a fullword alternative to MAXLENGTH.

MAXLENGTH(*data-value*)

specifies the maximum amount (halfword binary value) of data that CICS is to recover in response to a CONVERSE command. If INTO is specified, MAXLENGTH overrides the use of TOLENGTH as an input to CICS. If SET is specified, MAXLENGTH provides a way for the program to limit the amount of data it receives at one time.

If the value specified is less than zero, zero is assumed.

¹ Documented in the *IBM 3270 Data Stream Programmer's Reference* manual.

If the length of data exceeds the value specified and the NOTRUNCATE option is not present, the data is truncated to that value and the LENGERR condition occurs. The data area specified in the TOLENGTH option is set to the original length of data.

If the length of data exceeds the value specified and the NOTRUNCATE option is present, CICS retains the remaining data and uses it to satisfy subsequent RECEIVE commands. The data area specified in the TOLENGTH option is set to the length of data returned.

If no argument is coded for MAXLENGTH, CICS defaults to TOLENGTH.

NOTRUNCATE

specifies that, when the data available exceeds the length requested, the remaining data is not to be discarded but retained for retrieval by subsequent RECEIVE commands.

PSEUDOBIN

specifies that the data being read and written is to be translated from System/7 pseudobinary representation to hexadecimal.

SESSION(*name*)

specifies the symbolic identifier (1–4 characters) of a session TCTTE. This option specifies the alternate facility to be used. If both this option and CONVID are omitted, the principal facility for the task is used.

SET(*ptr-ref*)

specifies a pointer reference to be set to the address of data received from the conversation partner in an MRO conversation. The pointer reference, unless changed by other commands or statements, is valid until the next CONVERSE (MRO) command or the end of task.

If DATALOCATION(ANY) is associated with the application program, the address of the data can be above or below the 16MB line.

If DATALOCATION(BELOW) is associated with the application program, and the data resides above the 16MB line, the data is copied below the 16MB line, and the address of this copy is returned.

If TASKDATAKEY(USER) is specified for the running task, and storage protection is active, the data returned is in a user-key. If TASKDATAKEY(CICS) is specified and storage protection is active, the data returned is in a CICS-key.

STATE(*cvda*)

gets the state of the transaction program. The cvda values returned by CICS are:

- ALLOCATED
- FREE
- PENDFREE
- RECEIVE
- ROLLBACK
- SEND
- SYNCFREE

SYNCRECEIVE
SYNCSEND

STRFIELD

specifies that the data area specified in the FROM option contains structured fields. If this option is specified, the contents of all structured fields must be handled by the application program. The CONVERSE command, rather than a SEND command, must be used if the data area contains a read partition structured field. (Structured fields are described in the *CICS 3270 Data Stream Device Guide*.)

CTLCHAR and ERASE are mutually exclusive with STRFIELD, and their use with STRFIELD generates an error message.

TOFLENGTH(*data-area*)

is a fullword alternative to TOLENGTH.

TOLENGTH(*data-area*)

specifies the length, as a halfword binary value, of the data to be received. If you specify INTO, but omit MAXLENGTH, “data-area” specifies the maximum length that the program accepts. If the value is less than zero, zero is assumed.

If the length of the data exceeds the value specified, but NOTRUNCATE is omitted, the data is truncated to that value, and the LENGERR condition occurs. When the data is received, the data area is set to the length of the data.

For a description of a safe upper limit, see “LENGTH options” on page 5.

Conditions

Some of the following conditions can occur in combination with others. CICS checks for these conditions in the following order:

1. INBFMH
2. EOC

If more than one occurs, only the first is passed to the application program. EIBRCODE, however, is set to indicate all the conditions that occurred.

CBIDERR

occurs if the requested attach header control block named in ATTACHID cannot be found.

Default action: terminate the task abnormally.

EOC

occurs when a request/response unit (RU) is received with the end-of-chain indicator set. Field EIBEOC also contains this indicator.

Default action: ignore the condition.

EOF

occurs when an end-of-file indicator is received.

Default action: terminate the task abnormally.

CONVERSE (non-VTAM)

INBFMH

occurs if a request/response unit (RU) contains a function management header (FMH). Field EIBFMH contains this indicator and it should be used in preference to INBFMH. The IGNORE CONDITION command can be used to ignore the condition.

Default action: terminate the task abnormally.

LENGERR

occurs in any of the following situations:

- Data is discarded by CICS because its length exceeds the maximum that the program accepts and the NOTRUNCATE option is not specified.
- An out-of-range value is supplied in the FROMLENGTH option.

Default action: terminate the task abnormally.

NOTALLOC

occurs if the facility specified in the command is not owned by the application.

Default action: terminate the task abnormally.

RDATT

occurs if the “receive” part of the conversation is terminated by the attention (ATTN) key rather than the return key.

Default action: ignore the condition.

TERMERR

occurs for a session-related error.

A CANCEL TASK request by a user node error program (NEP) may cause this condition if the task has an outstanding terminal control request active when the node abnormal condition program handles the session error.

Default action: terminate the task abnormally with abend code ATNI.

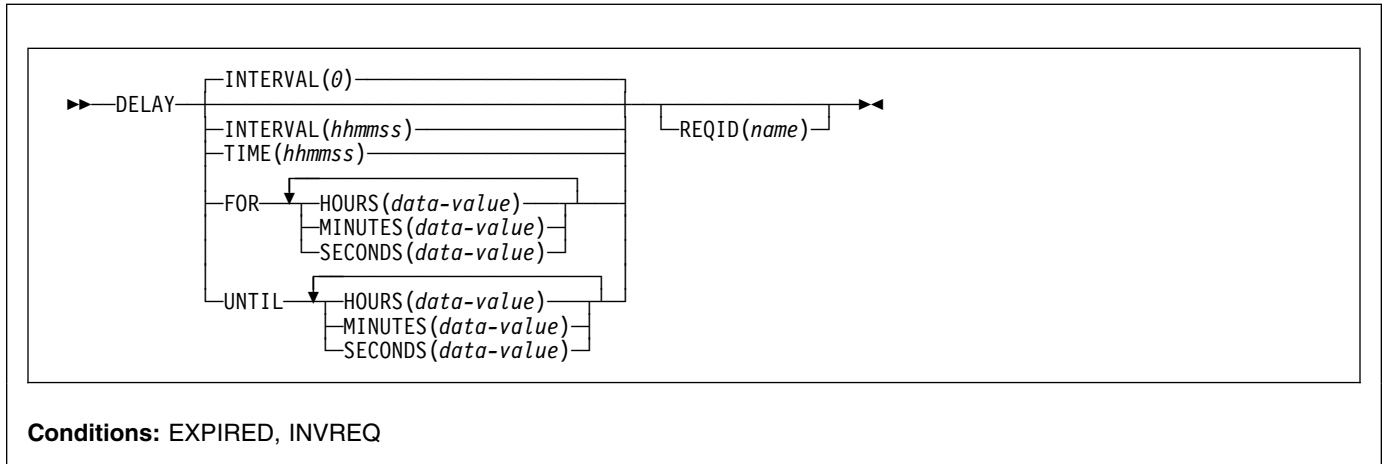
WRBRK

occurs if the “send” part of the conversation is terminated by the attention (ATTN) key rather than the return key.

Default action: ignore the condition.

DELAY

Delay the processing of a task.



Note for dynamic transaction routing

Using DELAY with REQID if later CANCELED could create inter-transaction affinities that adversely affect the use of dynamic transaction routing. See the *CICS Application Programming Guide* for more information about transaction affinities.

Description

DELAY suspends the processing of the issuing task for a specified interval of time or until a specified time of day. It supersedes any previously initiated POST command for the task.

The default is INTERVAL(0), but for C the default is FOR HOURS(0) MINUTES(0) SECONDS(0).

Options

FOR

specifies the duration of the delay.

HOURS(*data-value*)

a fullword binary value in the range 0–99.

INTERVAL(*hhmmss*)

specifies, in packed decimal format, the interval of time that is to elapse from the time when the DELAY command is issued. The **mm** and **ss** are in the range 0–59. The time specified is added to the current clock time by CICS when the command is executed to calculate the expiration time.

When using the C language, you are recommended to use the FOR/UNTIL HOURS, MINUTES, and SECONDS options as C does not provide a packed decimal data type. You may use INTERVAL, but if the value specified is **not** an integer constant, the application is responsible

for ensuring that the value passed to CICS is in packed decimal format.

MINUTES(*data-value*)

specifies a fullword binary value in the range 0–59, when HOURS or SECONDS are also specified, or 0–5999 when MINUTES is the only option specified.

REQID(*name*)

specifies a name (1–8 characters), which should be unique, to identify the DELAY request. Using this option to specify an application-defined name enables another transaction to cancel the DELAY request.

To enable other tasks to cancel unexpired DELAY requests, you must make the request identifier dynamically available. For example, storing it in a TS queue, whose name is known to other applications that may want to cancel the DELAY request, is one way you can pass a request identifier to other transactions.

SECONDS(*data-value*)

specifies a fullword binary value in the range 0–59, when HOURS or MINUTES are also specified, or 0–359 999 when SECONDS is the only option specified.

TIME(*hhmmss*)

specifies, in packed decimal format, the time when the task should resume processing.

When using the C language, you are recommended to use the FOR/UNTIL HOURS, MINUTES, and SECONDS options as C does not provide a packed decimal data

DELAY

type. You may use TIME, but if the value specified is **not** an integer constant, the application is responsible for ensuring that the value passed to CICS is in packed decimal format. See the section about expiration times in the *CICS Application Programming Guide*.

SECONDS(3723) means one hour, two minutes, and three seconds.

UNTIL

specifies the time at the end of the delay and when the task should resume processing.

Conditions

EXPIRED

occurs if the time specified has already expired when the command is issued.

Default action: ignore the condition.

INVREQ

RESP2 values:

- 4 Hours are out of range.
- 5 Minutes are out of range.
- 6 Seconds are out of range.

also occurs (RESP2 not set) if the DELAY command is not valid for processing by CICS.

Default action: terminate the task abnormally.

Examples

The following example shows you how to suspend the processing of a task for five minutes:

```
EXEC CICS DELAY
  INTERVAL(500)
  REQID('GXLBZQMR')
```

```
EXEC CICS DELAY FOR MINUTES(5)
```

The following example shows you how, at 09:00, to suspend the processing of a task until 12:45:

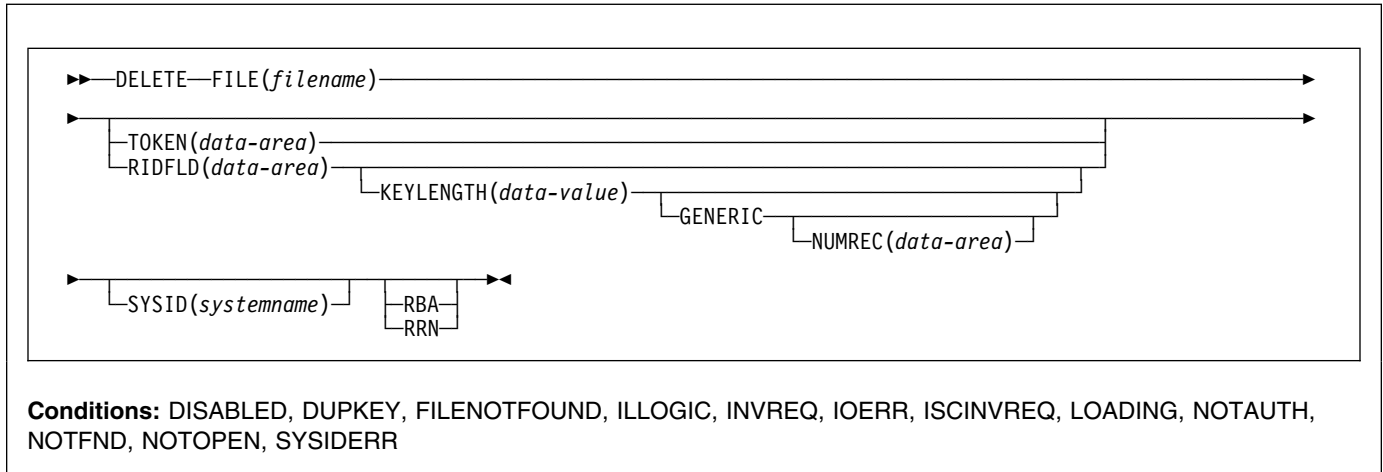
```
EXEC CICS DELAY
  TIME(124500)
  REQID('UNIQUECODE')
```

There are two ways to enter the time under FOR or UNTIL.

- A combination of at least two of HOURS(0–99), MINUTES(0–59) and SECONDS(0–59). HOURS(1) SECONDS(3) would mean one hour and three seconds (the minutes default to zero).
- Any one of HOURS(0–99), MINUTES(0–5999), or SECONDS(0–359999). HOURS(1) means one hour. MINUTES(62) means one hour and two minutes.

DELETE

Delete a record from a file — VSAM KSDS, VSAM RRDS, and data tables only.



Description

DELETE deletes a record from a file on a KSDS, a path over a KSDS, a CICS or user-maintained data table, or an RRDS. You cannot delete from a VSAM ESDS or a DAM data set.

All references to KSDS apply equally to CICS maintained data tables and, except where stated otherwise, to paths over a KSDS. The file may be on a local or a remote system. You identify, in the RIDFLD option, the specific record to be deleted.

You can delete a group of records in a similar way with a single invocation of this command, identifying the group by the GENERIC option (not available for RRDS).

You can also use this command to delete a single record that has previously been retrieved for update (by a READ UPDATE command). In this case, you must not specify the RIDFLD option.

When this command is used to delete records from a CICS-maintained data table, the update is made to both the source VSAM KSDS data set and the in-memory data table.

When this command is used to delete records from a user-maintained data table, the update is made only to the in-memory data table.

Options

FILE(filename)

specifies the name of the file to be accessed.

If SYSID is specified, the data set to which this file refers is assumed to be on a remote system irrespective of whether the name is defined in the FCT. Otherwise, the FCT entry is used to find out whether the data set is on a local or a remote system.

GENERIC (VSAM KSDS only)

specifies that the search key is a generic key with a length specified in the KEYLENGTH option. The search for a record is satisfied when a record is found with a key that has the same starting characters (generic key) as those specified.

KEYLENGTH(data-value)

specifies the length (halfword binary) of the key that has been specified in the RIDFLD option, except when RBA or RRN is specified, in which case it is not valid. This option must be specified if GENERIC is specified, and it may be specified whenever a key is specified. However, if the length specified is different from the length defined for the data set and the operation is not generic, the INVREQ condition occurs.

The INVREQ condition also occurs if you specify GENERIC, and the KEYLENGTH is not less than that specified in the VSAM definition.

You should not specify a zero value of KEYLENGTH because the results of this are unpredictable.

NUMREC(data-area) (VSAM KSDS only)

specifies a halfword binary data area that CICS sets to the number of deleted records.

DELETE

RBA (*VSAM KSDS base data sets only, not paths*)

specifies that the record identification field specified in the RIDFLD option contains a relative byte address. This option should be used only when deleting records using relative byte addresses instead of keys to identify the records.

You cannot use RBA for:

- User-maintained data tables
- KSDS files that are capable of holding more than 4GB of data

RIDFLD(*data-area*)

specifies the record identification field. The contents can be a key, a relative byte address, or a relative record number. For a relative byte address or a relative record number, the format of this field must be fullword binary. For a relative byte address, the RIDFLD can be greater than or equal to zero. For a relative record number, the RIDFLD can be greater than or equal to 1.

The contents must be a key for user-maintained data tables.

You must specify this option if you have also specified GENERIC.

RRN (*VSAM RRDS only*)

specifies that the record identification field specified in the RIDFLD option contains a relative record number. This option should be used only with files referencing relative record data sets.

SYSID(*systemname*)

specifies the name (1–4 characters) of the system the request is directed to.

If you specify SYSID, and omit both RBA and RRN, you must also specify KEYLENGTH; it cannot be found in the FCT.

TOKEN(*data-area*)

Specifies, as a fullword binary value, a unique identifier for this DELETE request. Use this to associate the delete request with a record returned on a previous READ UPDATE request. The value to use is the value returned in the TOKEN held by the earlier READ UPDATE request.

Conditions

Note: RESP2 values are not set for files that are on remote systems.

DISABLED

RESP2 values:

- 50** A file is disabled. A file may be disabled because:
- It was initially defined as disabled and has not since been enabled.
 - It has been disabled by a SET FILE or a CEMT SET FILE command.

This condition cannot occur when the DELETE follows any read with the UPDATE option.

Default action: terminate the task abnormally.

DUPKEY

RESP2 values:

- 140** A record is accessed by way of an alternate index with the NONUNIQUEKEY attribute, and another alternate index record with the same key follows.

Default action: terminate the task abnormally.

FILENOTFOUND

RESP2 values:

- 1** The file name referred to in the FILE option cannot be found in the file resource definition.

Default action: terminate the task abnormally.

ILLOGIC

RESP2 values:

- 110** A VSAM error occurs that does not fall within one of the other CICS response categories.

(See EIBRCODE in the EXEC interface block; refer to Appendix A, "EXEC interface block" on page 373 for details.)

Default action: terminate the task abnormally.

INVREQ

RESP2 values:

- 20** Delete operations are not allowed according to the file entry specification in the FCT.
- 21** A DELETE command is issued for a file referring to a VSAM ESDS. a VSAM ESDS or a DAM data set.
- 22** A generic delete is issued for a file that is not a VSAM KSDS.
- 25** The KEYLENGTH and GENERIC options are specified, and the length specified in the KEYLENGTH option is greater than or equal to the length of a full key.
- 26** The KEYLENGTH option is specified (but the GENERIC option is not specified), and the specified length does not equal the length defined for the data set to which this file refers.
- 27** A DELETE command is issued for a file referring to a DAM data set.
- 31** A DELETE command without the RIDFLD option is issued for a file for which no previous READ UPDATE command has been issued.
- 42** The KEYLENGTH and GENERIC options are specified, and the length specified in the KEYLENGTH option is less than zero.

- 44 The DELETE command does not conform to the format of DELETE for a user-maintained data table; for example if RBA was specified.
- 47 A DELETE instruction includes a token whose value cannot be matched against any token in use for an existing read for UPDATE request.
- 48 An attempt is made to function-ship a request that includes a TOKEN keyword.
- 49 An attempt was made to delete a file that is an invalid file type.

Default action: terminate the task abnormally.

IOERR

RESP2 values:

- 120 There is an I/O error during the file control operation. An I/O error is any unusual event that is not covered by a CICS condition.

For VSAM files, IOERR normally indicates a hardware error.

(Further information is available in the EXEC interface block; refer to Appendix A, "EXEC interface block" on page 373 for details.)

Default action: terminate the task abnormally.

ISCVREQ

RESP2 values:

- 70 The remote system indicates a failure that does not correspond to a known condition.

Default action: terminate the task abnormally.

LOADING

RESP2 values:

- 104 A delete request is issued for a user-maintained data table that is currently being loaded.

A user-maintained data table cannot be modified during loading.

Default action: terminate the task abnormally.

NOTAUTH

RESP2 values:

- 101 A resource security check has failed on FILE(filename).

Default action: terminate the task abnormally.

NOTFND

RESP2 values:

- 80 An attempt to delete a record based on the search argument provided is unsuccessful.

For user-maintained data tables, this condition occurs if an attempt to delete a record is unsuccessful because there is no entry with the specified key in the data table. This can occur on an attempt to delete a record using a DELETE

without RIDFLD, if the delete is associated with a READ UPDATE request for a record that this transaction has deleted (using DELETE with RIDFLD) after it was read for update.

This does not mean that there is no such record in the source data set; it may be that such a record is present but was either rejected during initial loading by the user exit XDTRD, or was subsequently deleted from the user-maintained data table.

Default action: terminate the task abnormally.

NOTOPEN

RESP2 values:

- 60 The requested file is CLOSED and UNENABLED. The CLOSED, UNENABLED state is reached after a CLOSE request has been received against an OPEN ENABLED file and the file is no longer in use. This state can also be specified as the initial state by means of the FILSTAT parameter of the DFHFCT TYPE=FILE macro, or by defining a file using the RDO options STATUS(UNENABLED) and OPENTIME(FIRSTREF).

The requested file is OPEN and in use by other transactions, but a CLOSE request against the file has been received.

This condition does not occur if the request is made to either a CLOSED, ENABLED file or a CLOSED, DISABLED file. In the first case, the file is opened as part of executing the request. In the second case, the DISABLED condition occurs.

This condition also cannot occur when deleting a record just read for update.

Default action: terminate the task abnormally.

SYSIDERR

RESP2 values:

- 130 The SYSID option specifies a name that is neither the local system nor a remote system (made known to CICS by defining a CONNECTION). SYSIDERR also occurs when the link to the remote system is closed.

Default action: terminate the task abnormally.

Examples

The following example shows you how to delete a group of records in a VSAM data set:

DELETE

```
EXEC CICS DELETE  
  FILE('MSTVSAM')  
  RIDFLD(ACCTNO)  
  KEYLENGTH(1en)  
  GENERIC  
  NUMREC(NUMDEL)
```

DELETEQ TD

Delete all transient data.

```

▶▶—DELETEQ TD—QUEUE(name)—└─SYSID(systemname)—◀◀

```

Conditions: DISABLED, INVREQ, ISCVREQ, NOTAUTH, QIDERR, SYSIDERR

Description

DELETEQ TD deletes all the transient data associated with a particular intrapartition destination (queue). All storage associated with the destination is released (deallocated). Note that you cannot use this command to delete an **extrapartition** transient data queue. An attempt to do so results in an INVREQ condition.

Options

QUEUE(*name*)

specifies the symbolic name (1–4 alphanumeric characters) of the queue to be deleted. The named queue must have been defined in the DCT.

If SYSID is specified, the queue is assumed to be on a remote system, whether or not the name is defined in the DCT. Otherwise the entry in the DCT is used to find out whether the queue is on a local or a remote system.

SYSID(*systemname*)

(remote systems only) specifies the name (1–4 characters) of the system the request is directed.

Conditions

DISABLED

occurs when the queue has been disabled.

Default action: terminate the task abnormally.

INVREQ

occurs if DELETEQ names an extrapartition queue.

Default action: terminate the task abnormally.

ISCVREQ

occurs when the remote system indicates a failure that does not correspond to a known condition.

Default action: terminate the task abnormally.

NOTAUTH

occurs when a resource security check has failed on QUEUE(*name*).

Default action: terminate the task abnormally.

QIDERR

occurs if the symbolic destination to be used with DELETEQ TD cannot be found.

Default action: terminate the task abnormally.

SYSIDERR

occurs when the SYSID option specifies a name that is neither the local system nor a remote system (made known to CICS by defining an RDO CONNECTION). SYSIDERR also occurs when the link to the remote system is closed.

Default action: terminate the task abnormally.

DELETEQ TS

Delete a temporary storage queue.

```

▶▶—DELETEQ TS—QUEUE(name)—SYSID(systemname)—◀◀

```

Conditions: INVREQ, ISCVREQ, NOTAUTH, QIDERR, SYSIDERR

Note for dynamic transaction routing

Using this command could create inter-transaction affinities that adversely affect the use of dynamic transaction routing. See the *CICS Application Programming Guide* for more information about transaction affinities.

Description

DELETEQ TS deletes all the temporary data associated with a temporary storage queue. All storage associated with the queue is freed.

You should delete temporary data as soon as possible to avoid using excessive amounts of storage.

When a recoverable temporary storage queue is deleted, you must issue a syncpoint before issuing a subsequent WRITEQ TS for the same queue.

Options

QUEUE(*name*)

specifies the symbolic name (1–8 alphanumeric characters) of the queue to be deleted. The name may not consist solely of binary zeros and must be unique within the CICS system.

SYSID(*systemname*)

(remote and shared queues only) specifies the system name (1–4 characters) identifying the remote system or shared queue pool to which the request is directed.

Conditions

INVREQ

occurs in either of the following situations:

- the queue was created by CICS internal code.
- the queue name specified consists solely of binary zeroes.

Default action: terminate the task abnormally.

ISCVREQ

occurs when the remote system indicates a failure that does not correspond to a known condition.

Default action: terminate the task abnormally.

NOTAUTH

occurs when a resource security check has failed on QUEUE(*name*).

Default action: terminate the task abnormally.

QIDERR

occurs when the specified queue cannot be found in either main or auxiliary storage.

Default action: terminate the task abnormally.

SYSIDERR

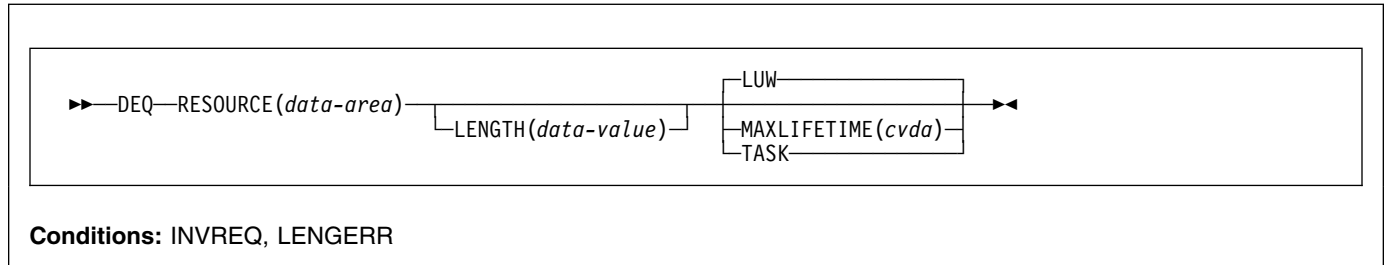
occurs in any of the following situations:

- When the SYSID option specifies a name that is neither the local system nor a remote system (made known to CICS by defining an RDO CONNECTION)
- When the link to the remote system is closed

Default action: terminate the task abnormally.

DEQ

Schedule use of a resource by a task (dequeue).



Note for dynamic transaction routing

Using this command could create inter-transaction affinities that adversely affect the use of dynamic transaction routing. See the *CICS Application Programming Guide* for more information about transaction affinities.

Description

DEQ causes a resource currently enqueued on by the task to be released for use by other tasks.

If a task enqueues on, but does not dequeue from, a resource, CICS automatically releases the resource during syncpoint processing or when the task is terminated. A resource in the context of this command is any string of 1–255 bytes, established by in-house standards, to protect against conflicting actions between tasks, or to cause single-threading within a program.

When issuing the DEQ command, the resource to be dequeued from must be identified by the method used when enqueueing on the resource. If no enqueue has been issued for the resource, the dequeue is ignored.

If more than one ENQ command is issued for a resource by a task, that resource remains owned by the task until the task issues a matching number of DEQ commands.

Options

LENGTH(*data-value*)

specifies that the resource to be dequeued from has a length given by the data value. The data value is a halfword binary value in the range 1 through 255. If the value you specify is outside this range, a LENGERR condition occurs. If the LENGTH option is specified in an ENQ command, it must also be specified in the DEQ command for that resource, and the values of these options must be the same.

MAXLIFETIME(*cvda*)

specifies the duration of the ENQ being released. CVDA values are:

LUW ENQ was acquired with a duration of a unit of work. This is the default value.

TASK ENQ was acquired with a duration of a task.

RESOURCE(*data-area*)

specifies either an area whose address represents the resource to be dequeued from, or a variable that contains the resource (an employee name, for example). In the latter case, you must use the LENGTH option.

Conditions

INVREQ

RESP2 values:

- 2 The MAXLIFETIME option is set with an incorrect CVDA.

Default action: terminate the task abnormally.

LENGERR

RESP2 values:

- 1 The value you specified for the LENGTH option is outside the range 1 through 255.

Default action: terminate the task abnormally.

DEQ

Examples

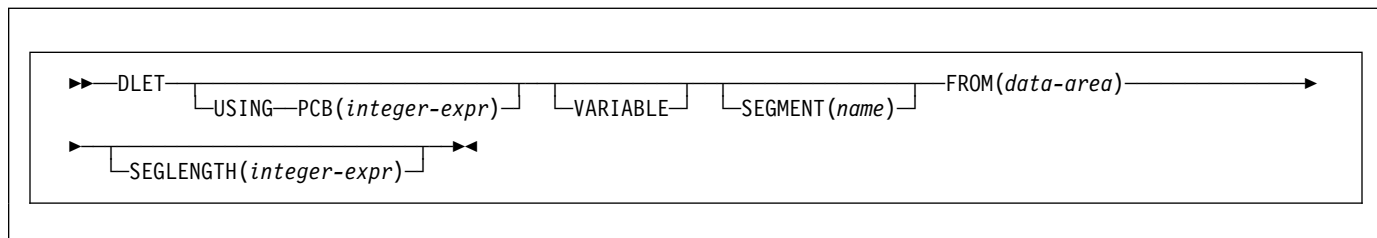
The following examples show how to dequeue from a resource:

```
EXEC CICS DEQ  
  RESOURCE(RESNAME)
```

```
EXEC CICS DEQ  
  RESOURCE(SOCSECNO)  
  LENGTH(8)
```

DLET (DL/I)

Delete a segment.



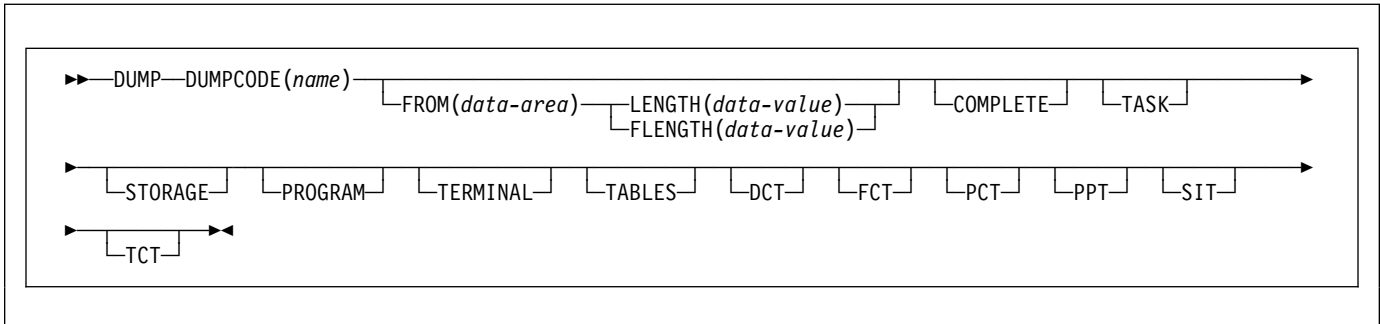
Description

The full form of the command is DELETE. See *DL/I DOS/VS Application Programming: High Level Programming Interfaces* for details.

DUMP

DUMP

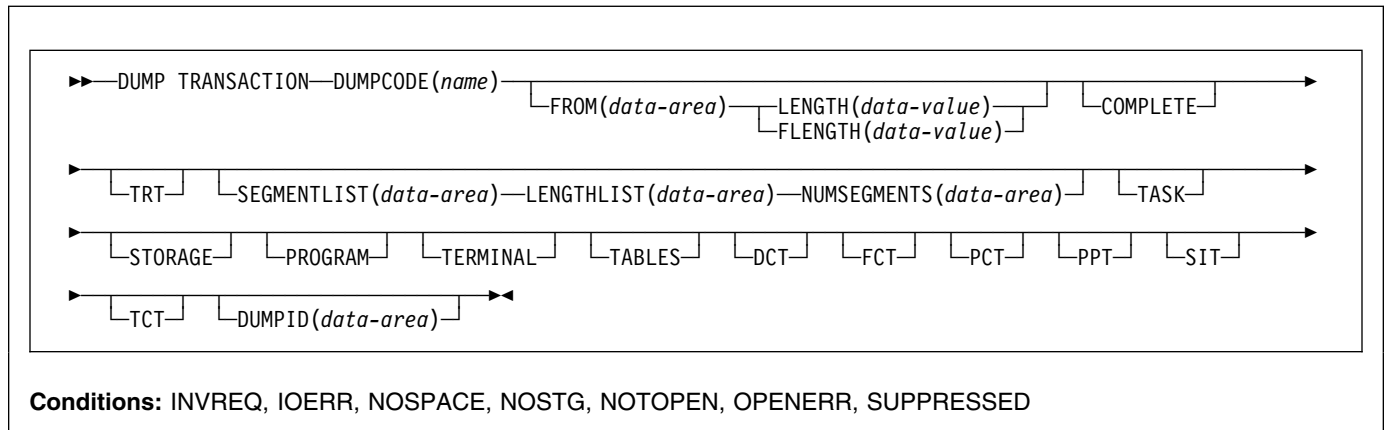
Request a transaction dump.



This command is now superseded by DUMP TRANSACTION, but is still supported for compatibility with previous releases.

DUMP TRANSACTION

Request a transaction dump.



Description

DUMP TRANSACTION dumps all, a series, or any single main storage area related to a task, any or all of the CICS tables (DCT, FCT, PCT, PPT, SIT, or TCT), or all of these together.

Options

COMPLETE

dumps all main storage areas related to a task and all the CICS tables.

DCT

dumps the destination control table.

DUMPCODE(*name*)

specifies a name (1–4 characters) that identifies the dump. If the name contains any leading or imbedded blanks, the dump is produced but INVREQ is raised. No entry is added to the system dump table.

If you omit all the options except DUMPCODE, you get the same dump as if you specified TASK.

DUMPID(*data-area*)

is a number returned by the system to identify the particular dump.

FCT

dumps the file control table.

FLENGTH(*data-value*)

specifies the length (fullword binary value) of the storage area (specified in the FROM option) that is to be dumped. The maximum length that you can specify is 16 777 215 bytes.

FLENGTH and LENGTH are mutually exclusive.

FROM(*data-area*)

dumps the specified data area, which must be a valid area; that is, storage allocated by the operating system within the CICS region. In addition, the following areas are dumped:

- Task control area (TCA) and, if applicable, the transaction work area (TWA).
- Common system area (CSA), including the user's portion of the CSA (CWA).
- If TRAN is specified for the TRTRAN system initialization parameter, only the trace entries associated with the current task are formatted. If TRTRAN=ALL is specified, the entire internal trace table is formatted. This applies only when the CICS trace facility is active.
- Either the terminal control table terminal entry (TCTTE) or the destination control table entry associated with the requesting task.

Whenever the TCTTE is dumped, the terminal control table user area (if any) and the message control blocks (if any) associated with the TCTTE are dumped. The latter are used by basic mapping support.

LENGTH(*data-value*)

specifies the length (halfword binary) of the data area specified in the FROM option. For a description of a safe upper limit, see "LENGTH options" on page 5.

LENGTH and FLENGTH are mutually exclusive.

LENGTHLIST(*data-area*)

specifies a list of 32-bit binary values showing the lengths of the storage areas to be dumped. This corresponds to the list of segments specified in the SEGMENTLIST option. You must use both the

DUMP TRANSACTION

SEGMENTLIST and NUMSEGMENTS options when you use the LENGTHLIST option.

NUMSEGMENTS(*data-area*)

specifies the number (fullword binary) of areas to be dumped. You must use both the SEGMENTLIST and LENGTHLIST options when you use the NUMSEGMENTS option.

PCT

formats a summary of each installed transaction definition.

PPT

dumps the processing program table.

PROGRAM

specifies that program storage areas associated with this task are to be dumped, as follows:

- Task control area (TCA) and, if applicable, the transaction work area (TWA)
- Common system area (CSA), including the user's portion of the CSA (CWA)
- Trace entries relating to the task written to the internal trace table (only when the CICS trace facility is active)
- All program storage areas containing user-written application programs active on behalf of the requesting task
- Register save areas (RSAs) indicated by the RSA chain off the TCA
- Either the terminal control table terminal entry (TCTTE) or the destination control table entry associated with the requesting task.

Whenever the TCTTE is dumped, the terminal control table user area (if any) and the message control blocks (if any) associated with the TCTTE are dumped.

SEGMENTLIST(*data-area*)

specifies a list of addresses, which are the starting points of the segments to be dumped. Each segment is a task-related storage area. You must use both the NUMSEGMENTS and LENGTHLIST options when you use the SEGMENTLIST option.

SIT

dumps the system initialization table.

STORAGE

specifies that storage areas associated with this task are to be dumped, as follows:

- Task control area (TCA) and, if applicable, the transaction work area (TWA)
- Common system area (CSA), including the user's portion of the CSA (CWA)
- Trace entries relating to the task written to the internal trace table (only when the CICS trace facility is active)

- All transaction storage areas
- Either the terminal control table terminal entry (TCTTE) or the destination control table entry associated with the requesting task

Whenever the TCTTE is dumped, the terminal control table user area (if any) and the message control blocks (if any) associated with the TCTTE are dumped.

TABLES

dumps the DCT, FCT, PCT, PPT, SIT, and the TCT.

TASK

specifies that storage areas associated with this task are to be dumped, as follows:

- A summary of the transaction environment associated with this task
- Common system area (CSA), including the user's portion of the CSA (CWA)
- Trace entries relating to the task written to the internal trace table (only when the CICS trace facility is active)
- All program storage areas containing user-written application programs active on behalf of the requesting task
- All transaction storage areas
- Either the terminal control table terminal entry (TCTTE) or the destination control table entry associated with the requesting task
- Register save areas (RSAs) indicated by the RSA chain off the TCA
- All terminal input/output areas (TIOAs) chained off the terminal control table terminal entry (TCTTE) for the terminal associated with the requesting task

Whenever the TCTTE is dumped, the terminal control table user area (if any) and the message control blocks (if any) associated with the TCTTE are dumped.

TCT

dumps the terminal control table.

TERMINAL

specifies that storage areas associated with the terminal are to be dumped, as follows:

- Task control area (TCA) and, if applicable, the transaction work area (TWA)
- Common system area (CSA), including the user's portion of the CSA (CWA)
- Trace entries relating to the task written to the internal trace table (only when the CICS trace facility is active)
- All terminal input/output areas (TIOAs) chained off the terminal control table terminal entry (TCTTE) for the terminal associated with the requesting task as

long as the request is not a write, or storage freezing is on for the task or the terminal

- Either the terminal control table terminal entry (TCTTE) or the destination control table entry associated with the requesting task.

Whenever the TCTTE is dumped, the terminal control table user area (if any) and the message control blocks (if any) associated with the TCTTE are dumped. The latter are used by basic mapping support.

TRT

.dumps the trace entries relating to the task written to the internal trace table.

Conditions

INVREQ

RESP2 values:

- 13** An incorrect DUMPCODE is specified. DUMPCODE contains unprintable characters, or leading or imbedded blanks.
- The dump is produced but no entry is added to the system dump table.

Default action: terminate the task abnormally.

IOERR

RESP2 values:

- 9** The SDUMP process is not authorized.
- 10** An error occurred during system dumping.
- 12** The SYSDUMP library is full.
- 13** The CICS routine issuing the SDUMP is unable to establish a recovery routine.

Default action: terminate the task abnormally.

NOSPACE

RESP2 values:

- 4** The transaction dump is incomplete due to lack of space.

Default action: terminate the task abnormally.

NOSTG

RESP2 values:

- 5** CICS has run out of working storage.

Default action: terminate the task abnormally.

NOTOPEN

RESP2 values:

- 6** The current CICS dump data set is not open.

Default action: terminate the task abnormally.

OPENERR

RESP2 values:

- 7** There is an error on opening, closing, or writing to the current CICS dump routine.

Default action: terminate the task abnormally.

SUPPRESSED

RESP2 values:

- 1** The transaction dump is suppressed by MAXIMUM in table.
- 2** The transaction dump is suppressed by NOTRANDUMP in table.
- 3** The transaction dump is suppressed by a user exit program.
- 8** The system dump is suppressed by the global action of the system initialization parameter DUMP.

Default action: terminate the task abnormally.

Examples

The following example shows how to request a dump of all the task-related storage areas, the terminal control table, and a specified data area:

```
EXEC CICS DUMP TRANSACTION
      DUMPCODE('name')
      FROM(data-area)
      LENGTH(data-value)
```

This second example (written in PL/I) shows you a case in which five task-related storage areas are dumped:

```
DCL storage_address(5)  POINTER,
     storage_length(5)  FIXED BIN(31),
     nsegs              FIXED BIN(31);
storage_address(1) = ADDR(areal);
storage_length(1)  = CSTG(areal);
:
nsegs = 5;
EXEC CICS DUMP TRANSACTION
      DUMPCODE('name')
      SEGMENTLIST(storage_address)
      LENGTHLIST(storage_length)
      NUMSEGMENTS(nsegs);
```

ENDBR

ENDBR

End browse of a file.

```
▶▶—ENDBR—FILE(filename)—└─REQID(data-value)—└─SYSID(systemname)—◀◀
```

Conditions: FILENOTFOUND, ILLOGIC, INVREQ, IOERR, ISCINVREQ, NOTAUTH, SYSIDERR

Description

ENDBR ends a browse on a file or data table (CICS or user-maintained) on a local or a remote system.

The UPDATE option is available within browse so we recommend that you use this because otherwise you would need to issue an ENDBR command before using READ UPDATE to avoid self deadlock abends. We recommend issuing an ENDBR before syncpoint for similar reasons.

If STARTBR was not successful, you need not issue ENDBR.

Options

FILE(*filename*)

specifies the name of the file being browsed.

If SYSID is specified, the data set to which this file refers is assumed to be on a remote system irrespective of whether the name is defined in the FCT. Otherwise, the FCT entry is used to find out whether the data set is on a local or a remote system.

REQID(*data-value*)

specifies a unique (halfword binary value) request identifier for a browse, used to control multiple browse operations on a data set. If this option is not specified, a default value of zero is assumed.

SYSID(*systemname*)

specifies the name (1–4 characters) of the system the request is directed to.

Conditions

Note: RESP2 values are not set for files that are on remote systems.

FILENOTFOUND

RESP2 values:

- 1 The name referred to in the FILE option cannot be found in the FCT.

Default action: terminate the task abnormally.

ILLOGIC (VSAM)

RESP2 values:

- 110 A VSAM error occurs that does not fall within one of the other CICS response categories.
(See EIBRCODE in the EXEC interface block; refer to Appendix A, “EXEC interface block” on page 373 for details.)

Default action: terminate the task abnormally.

INVREQ

RESP2 values:

- 35 The REQID, if any, does not match that of any successful STARTBR command.
- # 49 An attempt was made to browse a file that is an invalid file type.

Default action: terminate the task abnormally.

IOERR

RESP2 values:

- 120 There is an I/O error during the file control operation. An I/O error is any unusual event that is not covered by a CICS condition.
For VSAM files, IOERR normally indicates a hardware error.
(Further information is available in the EXEC interface block; see Appendix A, “EXEC interface block” on page 373 for details.)

Default action: terminate the task abnormally.

ISCINVREQ

RESP2 values:

- 70 The remote system indicates a failure that does not correspond to a known condition.

Default action: terminate the task abnormally.

NOTAUTH

RESP2 values:

101 A resource security check has failed on FILE(filename).

Default action: terminate the task abnormally.

SYSIDERR

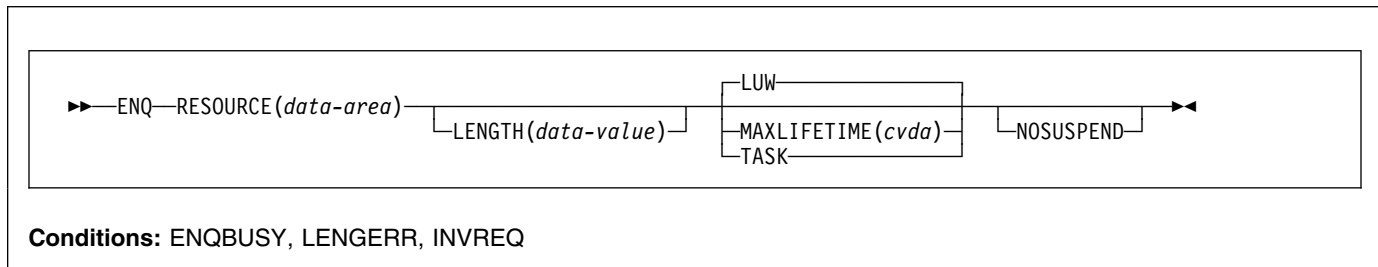
RESP2 values:

130 The SYSID option specifies a name that is neither the local system nor a remote system (made known to CICS by defining an RDO CONNECTION). SYSIDERR also occurs when the link to the remote system is closed.

Default action: terminate the task abnormally.

ENQ

Schedule use of a resource by a task (enqueue).



Note for dynamic transaction routing

Using this command could create inter-transaction affinities that adversely affect the use of dynamic transaction routing. See the *CICS Application Programming Guide* for more information about transaction affinities.

Description

ENQ causes further execution of the task issuing the ENQ command to be synchronized with the availability of the specified resource; control is returned to the task when the resource is available.

A resource in the context of this command is any string of 1–255 bytes, established by in-house standards, to protect against conflicting actions between tasks, or to cause single threading within a program.

If a task enqueues on a resource but does not dequeue from it, CICS automatically releases the resource during syncpoint processing (including DL/I, PCB, and TERM calls), or when the task is terminated. Option LUW forces the dequeue at the end of a unit of work (LUW). Option TASK forces the dequeue at the end of a task. If there are several units of work in a task, the enqueue carries over the LUWs.

If more than one ENQ command is issued for the same resource by a given task, the resource remains owned by that task until the task issues a matching number of DEQ commands.

The resource to be enqueued on must be identified by one of the following methods:

- Specifying a data area that is the resource. It is the location (address) of the data area in storage that matters, not its contents.
- Specifying a data area that contains a unique argument (for example, an employee name) that represents the resource. It is the contents of the data value that matters, not its location.

Resource unavailability

If a resource is not available when ENQUEUED, the application program is suspended until it becomes available. However, if the NOSUSPEND option has been specified and the resource is unavailable, the ENQBUSY condition is raised, as it is also raised if you have an active HANDLE condition. This allows the application program to handle the case of resource unavailability (by HANDLE CONDITION ENQBUSY) without waiting for the resource to become available.

Options

LENGTH(*data-value*)

specifies as a halfword binary value the length of the resource to be enqueued on. The value must be in the range 1 through 255; otherwise, the LENGERR condition occurs. If the LENGTH option is specified in an ENQ command, it must also be specified in the DEQ command for that resource, and the values of these options must be the same. You must specify LENGTH when using the method that specifies a data value containing a unique argument, but not for the method that specifies a data area as the resource. It is the presence or absence of LENGTH that tells CICS which method you are using.

MAXLIFETIME(*cvda*)

specifies the duration of the ENQ to be automatically released by CICS. CVDA values are:

LUW The duration of the ENQ is a unit of work. Examples are a syncpoint rollback or syncpoint, if the application does not issue a DEQ before the unit of work ends. This is the default value.

TASK

The duration of the ENQ is a task. The enqueue carries over the units of work within the task. Use MAXLIFETIME(TASK) with great care because other tasks issuing ENQ commands on the same resource could be suspended until the end of this task.

There are two ways to code this option.

- You can assign a cvda value with the translator routine DFHVALUE. This allows you to change a cvda value in the program. For example:

```
MOVE DFHVALUE(LUW) TO AREA-A
EXEC CICS ENQ RESOURCE(RESNAME)
        MAXLIFETIME(AREA-A)
```

- If the required action is always the same, you can declare the value directly. For example:

```
EXEC CICS ENQ RESOURCE(RESNAME) LUW
```

or

```
EXEC CICS ENQ RESOURCE(RESNAME) TASK
```

NOSUSPEND

specifies that the application program is not to be suspended if the resource on the ENQ command is unavailable, but the ENQBUSY condition occurs.

Note, however, that if a HANDLE CONDITION for ENQBUSY is active when the command is executed, action, control is passed to the user label supplied in the HANDLE CONDITION. This takes precedence over the NOSUSPEND option but is, of course, negated by either NOHANDLE or RESP.

RESOURCE(data-area)

identifies the resource to be enqueued on by:

- Specifying an area whose address represents the resource.
- Specifying a variable that contains the resource (an employee name, for example). In this case, you must use the LENGTH option.

Conditions**ENQBUSY**

occurs when an ENQ command specifies a resource that is unavailable and the NOSUSPEND option is specified, or there is an active HANDLE CONDITION ENQBUSY.

If the NOSUSPEND option is not specified, and the ENQ command specifies a resource that is unavailable, the

application program is suspended and the ENQBUSY condition is not raised.

Default action: ignore the condition.

INVREQ

RESP2 values: CVDA values are:

- The MAXLIFETIME option is set with an incorrect CVDA.

Default action: terminate the task abnormally.

LENGERR

RESP2 values:

- The value specified for the LENGTH option is outside the range 1 through 255.

Default action: terminate the task abnormally.

Examples

Two tasks, enqueueing on the same resource and specifying a data area that is the resource, must refer to the same location in storage. They could both, for example, refer to the same location in the CWA.

```
EXEC CICS ENQ
        RESOURCE(RESNAME)
```

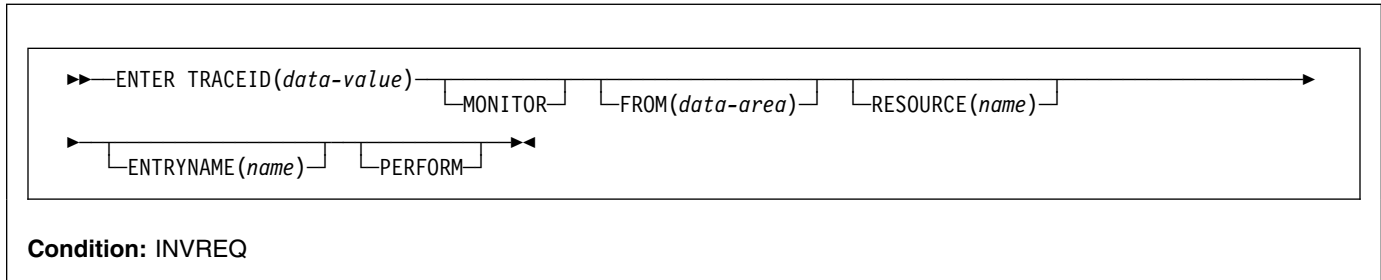
Two tasks, enqueueing on the same resource and specifying a data area that contains a unique argument, can refer to the same location or to different locations, but the contents of the locations must be the same. The length must be supplied in the LENGTH option.

```
EXEC CICS ENQ
        RESOURCE(SOCSECNO)
        LENGTH(8)
```

ENTER TRACEID

ENTER TRACEID

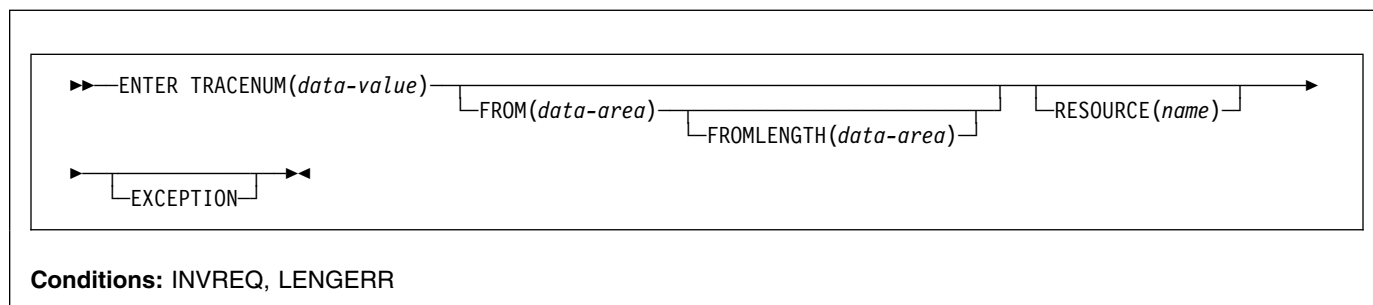
Specify a user trace entry point or an event monitoring point (EMP).



This command is now superseded by ENTER TRACENUM and MONITOR, but is still supported for compatibility with previous releases.

ENTER TRACENUM

Write a trace entry.



Description

The ENTER TRACENUM command makes a trace entry in the currently active trace destinations. CICS writes the trace entry only if the master and user trace flags are on, unless you specify the EXCEPTION option, in which case a user trace entry is always written, even if the master and user trace flags are off. Exception trace entries are always written to the internal trace table (even if internal tracing is set off), but they are written to other destinations only if they are active.

You can use the exception trace option in an application program to write a trace entry when the program detects an exception or abnormal condition. To do this, include an ENTER TRACENUM(data-value) EXCEPTION command in your program's exception or abnormal condition error-handling routine.

To write an exception trace entry in an error situation when an application program has given up control, you can issue an ENTER TRACENUM(data-value) EXCEPTION command from a user-written program error program (PEP). See the *CICS Customization Guide* for programming information about modifying the DFHPEP program.

Note ENTER TRACENUM replaces the earlier ENTER TRACEID command, which is still supported for compatibility with releases of CICS earlier than CICS/VSE® 2.3. You should use ENTER TRACENUM for all new programs, and whenever you apply maintenance to old programs.

For information about the trace entry format, see *CICS Problem Determination Guide*.

Options

EXCEPTION

specifies that CICS is to write a user exception trace entry. The EXCEPTION option overrides the master user trace flag, and CICS writes the trace entry even if the user trace flag is off. Exception trace entries are identified by the characters *EXCU when the trace entries are formatted by the trace utility program. See the *CICS Problem Determination Guide* for more information about user exception trace entries.

FROM(data-area)

specifies a data area (eight bytes) whose contents are to be entered into the data field of the trace table entry. If you omit the FROM option, two fullwords of binary zeros are passed.

FROMLENGTH(data-area)

specifies a halfword binary data area containing the length of the trace data, in the range 0–4000 bytes. If FROMLENGTH is not specified, a length of 8 bytes is assumed.

RESOURCE(name)

specifies an 8-character name to be entered into the resource field of the trace table entry.

TRACENUM(data-value)

specifies the trace identifier for a user trace table entry as a halfword binary value in the range 0 through 199.

Conditions

INVREQ

RESP2 values:

- 1 TRACENUM is outside the range 0 through 199.
- 2 There is no valid trace destination.
- 3 The user trace flag is set OFF and EXCEPTION has not been specified.

ENTER TRACENUM

Default action: terminate the task abnormally.

LENGERR

RESP2 values:

- 4 FROMLENGTH is outside the range 0 through 4000.

Default action: terminate the task abnormally.

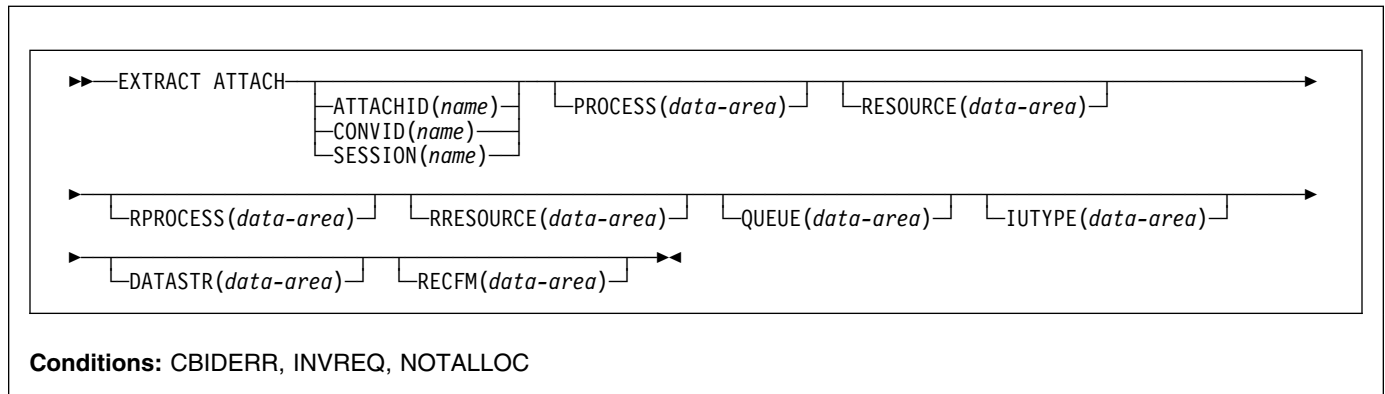
Examples

The following COBOL example shows how to write a user trace entry with a trace identifier of 123, with trace data from a data area called USER-TRACE-ENTRY:

```
EXEC CICS ENTER TRACENUM(123)
      FROM(USER-TRACE-ENTRY)
END-EXEC.
```

EXTRACT ATTACH (LUTYPE6.1)

Retrieve values from an LUTYPE6.1 attach header.



Description

EXTRACT ATTACH retrieves a set of values that are held in an attach header control block, or that have been built previously. For the command to retrieve information from a received attach Function Management Header (FMH), EIBATT must have been set during a RECEIVE or CONVERSE command.

Options

ATTACHID(*name*)

specifies that values are to be retrieved from an attach header control block. The name (1–8 characters) identifies this control block to the local task.

CONVID(*name*)

identifies the conversation to which the command relates. The 4-character name identifies either the token returned by a previously executed ALLOCATE command in EIBRSRCE in the EIB, or the token representing the principal session (returned by a previously executed ASSIGN command).

DATASTR(*data-area*)

corresponds to the data stream profile field, ATTDSP, in an LUTYPE6.1 attach FMH.

For communication between two CICS systems, no particular significance is given by CICS to the data stream profile field in an attach FMH. For most CICS applications, the option can be omitted.

The value returned in the data area is a halfword binary value. Only the low-order byte is used; the SNA-defined meanings of the bits are as follows:

0-7	reserved - must be set to zero
8-11	0000 - user-defined
	1111 - SCS data stream
	1110 - 3270 data stream
	1101 - structured field
	1100 - logical record management
12-15	defined by the user if bits 8-11 are set to 0000; otherwise reserved (must be set to zero)

IUTYPE(*data-area*)

corresponds to the interchange unit field, ATTIU, in an LUTYPE6.1 attach FMH.

For communication between two CICS systems, no particular significance is attached by CICS to the interchange unit field in an attach FMH. For most CICS applications the option can be omitted.

The value returned in the data area is a halfword binary value. Only the low-order 7 bits are used; the SNA-defined meanings of the bits are as follows:

0-10	reserved - must be set to zero
11	0 - not end of multichain interchange unit
	1 - end of multichain interchange unit
12,13	reserved - must be set to zero
14,15	00 - multichain interchange unit
	01 - single-chain interchange unit
	10 - reserved
	11 - reserved

PROCESS(*data-area*)

corresponds to the process name, ATDPN, in an LUTYPE6.1 attach FMH.

For communication between two CICS systems, a transaction running in one system can acquire a session

EXTRACT ATTACH (LUTYPE6.1)

to the second system and can identify the transaction to be attached; in the second system the identification is carried in the first chain of data sent across the session.

In general, the first four bytes of data identify the transaction to be attached. However an attach FMH, identifying the transaction to be attached, can be built and sent. The receiving CICS system uses just the first four bytes of the process name as a transaction name.

No significance is attached by CICS to process names in attach FMHs sent in chains of data other than the first.

For communication between a CICS system and another subsystem, refer to documentation supplied by the subsystem about how to use the process name field in an attach FMH.

QUEUE(*data-area*)

corresponds to the queue name, ATTDQN, in an attach FMH.

For communication between two CICS systems, no significance is attached by CICS to the queue name in an attach FMH.

For communication between a CICS system and another subsystem, refer to documentation supplied by the subsystem about how to use the queue name field in an attach FMH.

RECFM(*data-area*)

corresponds to the deblocking algorithm field, ATTDDBA, in an LUTYPE6.1 attach FMH.

For communication between two CICS systems, no particular significance is attached by CICS to the deblocking algorithm field in an attach FMH. For most CICS applications, the option can be omitted.

For communication between a CICS system and another subsystem, refer to documentation supplied by the subsystem about how to use the interchange unit field in an attach FMH.

The value returned in the data area is a halfword binary value. Only the low-order byte is used; the SNA-defined meanings of the bits are as follows:

0-7	reserved - must be set to zero
8-15	X'00' - reserved
	X'01' - variable-length variable-blocked
	X'02' - reserved
	X'03' - reserved
	X'04' - chain of RUs
	X'05' through X'FF' - reserved

RESOURCE(*data-area*)

corresponds to the resource name, ATTPRN, in an LUTYPE6.1 attach FMH.

For communication between two CICS systems, no significance is attached by CICS to the resource name in an attach FMH.

For communication between a CICS system and another subsystem, refer to documentation supplied by the subsystem about how to use the resource name field in an attach FMH.

RPROCESS(*data-area*)

corresponds to the return process name, ATTRDPN, in an LUTYPE6.1 attach FMH.

For communication between two CICS systems, no significance is attached by CICS to the return process name in an attach FMH.

For communication between a CICS system and another subsystem, refer to documentation supplied by the subsystem about how to use the return process name field in an attach FMH.

RRESOURCE(*data-area*)

corresponds to the return resource name, ATTRPRN, in an LUTYPE6.1 attach FMH.

For communication between two CICS systems, no significance is attached by CICS to the return resource name in an attach FMH.

For communication between a CICS system and another subsystem, refer to documentation supplied by the subsystem about how to use the return resource name field in an attach FMH.

SESSION(*name*)

specifies the symbolic identifier (1–4 characters) of a session TCTTE. This option specifies the alternate facility to be used.

Conditions

CBIDERR

occurs if the requested attach header control block cannot be found.

Default action: terminate the task abnormally.

INVREQ

occurs if incorrect data is found.

Default action: terminate the task abnormally.

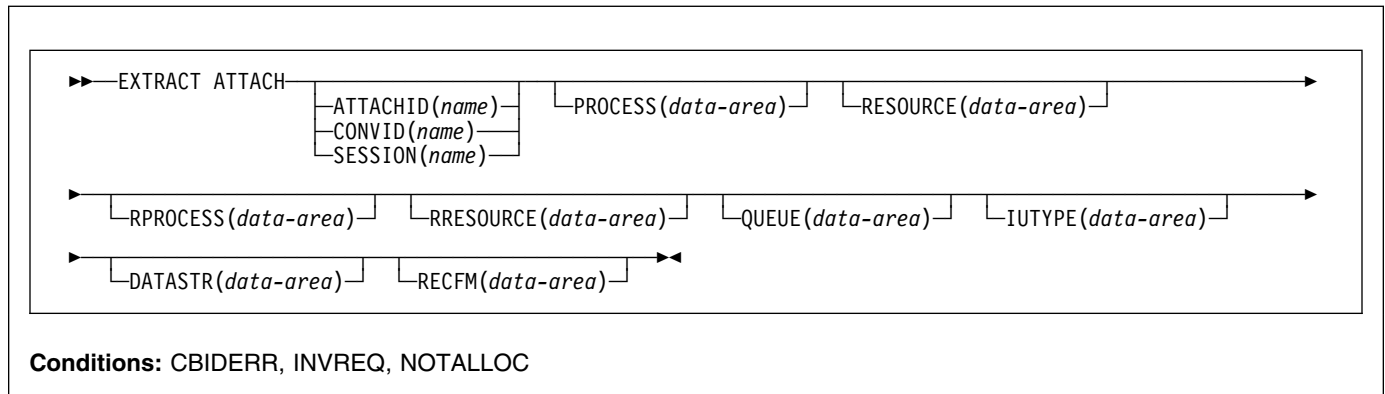
NOTALLOC

occurs if the facility specified in the command is not owned by the application.

Default action: terminate the task abnormally.

EXTRACT ATTACH (MRO)

Retrieve values from an MRO attach header.



Description

EXTRACT ATTACH retrieves a set of values that are held in an attach header control block, or that have been built previously. For the command to retrieve information from a received attach Function Management Header (FMH), EIBATT must have been set during a RECEIVE or CONVERSE command.

For more information about MRO and IRC, see the *CICS Intercommunication Guide*.

Options

ATTACHID(name)

specifies that values are to be retrieved from an attach header control block. The name (1–8 characters) identifies this control block to the local task.

CONVID(name)

identifies the conversation to which the command relates. The 4-character name identifies either the token returned by a previously executed ALLOCATE command in EIBRSRCE in the EIB, or the token representing the principal session (returned by a previously executed ASSIGN command).

DATASTR(data-area)

corresponds to the data stream profile field, ATTDSP, in an LUTYPE6.1 attach FMH.

For communication between two CICS systems, no particular significance is given by CICS to the data stream profile field in an attach FMH. For most CICS applications, the option can be omitted.

The value returned in the data area is a halfword binary value. Only the low-order byte is used; the SNA-defined meanings of the bits are as follows:

0-7	reserved - must be set to zero
8-11	0000 - user-defined
	1111 - SCS data stream
	1110 - 3270 data stream
	1101 - structured field
	1100 - logical record management
12-15	defined by the user if bits 8-11 are set to 0000; otherwise reserved (must be set to zero)

IUTYPE(data-area)

corresponds to the interchange unit field, ATTIU, in an LUTYPE6.1 attach FMH.

For communication between two CICS systems, no particular significance is attached by CICS to the interchange unit field in an attach FMH. For most CICS applications the option can be omitted.

The value returned in the data area is a halfword binary value. Only the low-order 7 bits are used; the SNA-defined meanings of the bits are as follows:

0-10	reserved - must be set to zero
11	0 - not end of multichain interchange unit
	1 - end of multichain interchange unit
12,13	reserved - must be set to zero
14,15	00 - multichain interchange unit
	01 - single chain interchange unit
	10 - reserved
	11 - reserved

PROCESS(data-area)

corresponds to the process name, ATTDPN, in an LUTYPE6.1 attach FMH.

For communication between two CICS systems, a transaction running in one system can acquire a session

EXTRACT ATTACH (MRO)

to the second system and can identify the transaction to be attached; in the second system the identification is carried in the first chain of data sent across the session.

In general, the first four bytes of data identify the transaction to be attached. However an attach FMH, identifying the transaction to be attached, can be built and sent. The receiving CICS system uses just the first four bytes of the process name as a transaction name.

No significance is attached by CICS to process names in attach FMHs sent in chains of data other than the first.

For communication between a CICS system and another subsystem, refer to documentation supplied by the subsystem about how to use the process name field in an attach FMH.

QUEUE(*data-area*)

corresponds to the queue name, ATTDQN, in an attach FMH.

For communication between two CICS systems, no significance is attached by CICS to the queue name in an attach FMH.

For communication between a CICS system and another subsystem, refer to documentation supplied by the subsystem about how to use the queue name field in an attach FMH.

RECFM(*data-area*)

corresponds to the deblocking algorithm field, ATTDDBA, in an LUTYPE6.1 attach FMH.

For communication between two CICS systems, no particular significance is attached by CICS to the deblocking algorithm field in an attach FMH. For most CICS applications, the option can be omitted.

For communication between a CICS system and another subsystem, refer to documentation supplied by the subsystem about how to use the interchange unit field in an attach FMH.

The value returned in the data area is a halfword binary value. Only the low-order byte is used; the SNA-defined meanings of the bits are as follows:

0-7	reserved - must be set to zero
8-15	X'00' - reserved
	X'01' - variable-length variable-blocked
	X'02' - reserved
	X'03' - reserved
	X'04' - chain of RUs
	X'05' through X'FF' - reserved

RESOURCE(*data-area*)

corresponds to the resource name, ATTPRN, in an LUTYPE6.1 attach FMH.

For communication between two CICS systems, no significance is attached by CICS to the resource name in an attach FMH.

For communication between a CICS system and another subsystem, refer to documentation supplied by the subsystem about how to use the resource name field in an attach FMH.

RPROCESS(*data-area*)

corresponds to the return process name, ATTRDPN, in an LUTYPE6.1 attach FMH.

For communication between two CICS systems, no significance is attached by CICS to the return process name in an attach FMH.

For communication between a CICS system and another subsystem, refer to documentation supplied by the subsystem about how to use the return process name field in an attach FMH.

RRESOURCE(*data-area*)

corresponds to the return resource name, ATTRPRN, in an LUTYPE6.1 attach FMH.

For communication between two CICS systems, no significance is attached by CICS to the return resource name in an attach FMH.

For communication between a CICS system and another subsystem, refer to documentation supplied by the subsystem about how to use the return resource name field in an attach FMH.

SESSION(*name*)

specifies the symbolic identifier (1–4 characters) of a session TCTTE. This option specifies the alternate facility to be used.

Conditions

CBIDERR

occurs if the requested attach header control block cannot be found.

Default action: terminate the task abnormally.

INVREQ

occurs if incorrect data is found.

Default action: terminate the task abnormally.

NOTALLOC

occurs if the facility specified in the command is not owned by the application.

Default action: terminate the task abnormally.

EXTRACT ATTRIBUTES (APPC)

Obtain the state of the APPC conversation.

```
▶▶—EXTRACT ATTRIBUTES—└─CONVID(name)—┘—STATE(cvda)—▶▶
```

Conditions: INVREQ, NOTALLOC

Description

EXTRACT ATTRIBUTES extracts conversation state information for APPC mapped conversations.

Options

CONVID(*name*)

identifies the conversation to which the command relates. The 4-character name identifies either the token returned by a previously executed ALLOCATE command in EIBRSRCE in the EIB, or the token representing the principal session (returned by a previously executed ASSIGN command).

For compatibility with earlier releases, SESSION is accepted as a synonym for CONVID. New programs should use CONVID.

By default, the principal facility is assumed.

STATE(*cvda*)

gets the state of the transaction program. The *cvda* values returned by CICS are:

```
ALLOCATED
CONFFREE
CONFRECEIVE
CONFSEND
FREE
PENDFREE
PENDRECEIVE
RECEIVE
ROLLBACK
SEND
SYNCFREE
SYNCRECEIVE
SYNCSEND
```

Conditions

INVREQ

RESP2 values:

- 200** A distributed program link server application explicitly, or implicitly by default, specified the function-shipping session (its principal facility) on the CONVID option.

also occurs (RESP2 not set) in any of the following situations:

- The command is issued against a CPI-Communications conversation.
- The command is issued against an APPC basic conversation. (A GDS EXTRACT ATTRIBUTES should have been used instead.)

Default action: terminate the task abnormally.

NOTALLOC

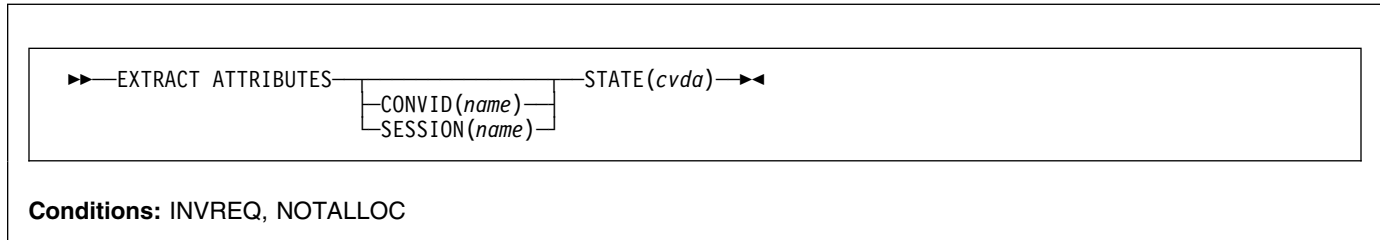
occurs if the specified CONVID value does not relate to a conversation owned by the application.

Default action: terminate the task abnormally.

EXTRACT ATTRIBUTES (MRO)

EXTRACT ATTRIBUTES (MRO)

Extract attributes from an MRO conversation.



Description

EXTRACT ATTRIBUTES (MRO) extracts conversation state information for MRO conversations.

Options

CONVID(*name*)

identifies the conversation to which the command relates. The 4-character name identifies either the token returned by a previously executed ALLOCATE command in EIBRSRCE in the EIB, or the token representing the principal session (returned by a previously executed ASSIGN command).

SESSION(*name*)

specifies the symbolic identifier (1–4 characters) of a session TCTTE. This option specifies the alternate facility to be used.

If both this option and CONVID are omitted, the principal facility for the task is used.

STATE(*cvda*)

gets the state of the transaction program. The *cvda* values returned by CICS are:

ALLOCATED
FREE
PENDFREE
RECEIVE
ROLLBACK
SEND
SYNCFREE
SYNCRECEIVE
SYNCSSEND

Conditions

INVREQ

RESP2 values:

- 200** A distributed program link server application explicitly, or implicitly by default, specified the function-shipping session (its principal facility) on the CONVID option.

also occurs (RESP2 not set) in any of the following situations:

- An incorrect command has been issued for the terminal or LU in use.

Default action: terminate the task abnormally.

NOTALLOC

occurs if the facility specified in the command is not owned by the application.

Default action: terminate the task abnormally.

EXTRACT LOGONMSG

Access VTAM logon data.

```

▶▶—EXTRACT LOGONMSG—INTO(data-area)—LENGTH(data-area)—▶▶
      |_____|
      |_____|
      SET(ptr-ref)
  
```

Condition: NOTALLOC

Description

EXTRACT LOGONMSG accesses VTAM logon data. This data may have been specified by the terminal operator at logon or in the ISSUE PASS command, for example. This data is only available if the system initialization parameter LGNMSG=YES is specified. The data can only be extracted once. It is possible to force the first transaction that runs on the terminal to be that which issues EXTRACT LOGONMSG by using the the system initialization parameter GMTRAN.

All the logon data is extracted and its length placed in the field specified by the LENGTH option. Because the LENGTH option cannot be used to limit the amount of data extracted, it is recommended that a field of 256 bytes is always used if INTO is specified.

If you use the SET option, the VTAM logon data is not freed until the session terminates (CLSDST). If you use the INTO option, the VTAM logon data is copied into user storage and then freed.

Options

INTO(*data-area*)

specifies the receiving field for the data extracted.

LENGTH(*data-area*)

specifies the length, as a halfword binary value, of the data extracted. If no data is available, LENGTH is set to zero.

SET(*ptr-ref*)

specifies the pointer reference that is to be set to the address of the data extracted. The pointer reference, unless changed by other commands or statements, is valid until the next EXTRACT LOGONMSG command or the end of task.

If DATALOCATION(ANY) is specified on the RDO PROGRAM resource definition for the application program, the address of the data can be above or below the 16MB line.

If DATALOCATION(BELOW) is specified, and the data resides above the 16MB line, the data is copied below the 16MB line, and the address of this copy is returned.

If TASKDATAKEY(USER) is specified on the RDO TRANSACTION resource definition for for the running task, and storage protection is active, the data returned is in a user-key. If TASKDATAKEY(CICS) is specified and storage protection is active, the data returned is in a CICS-key.

Conditions

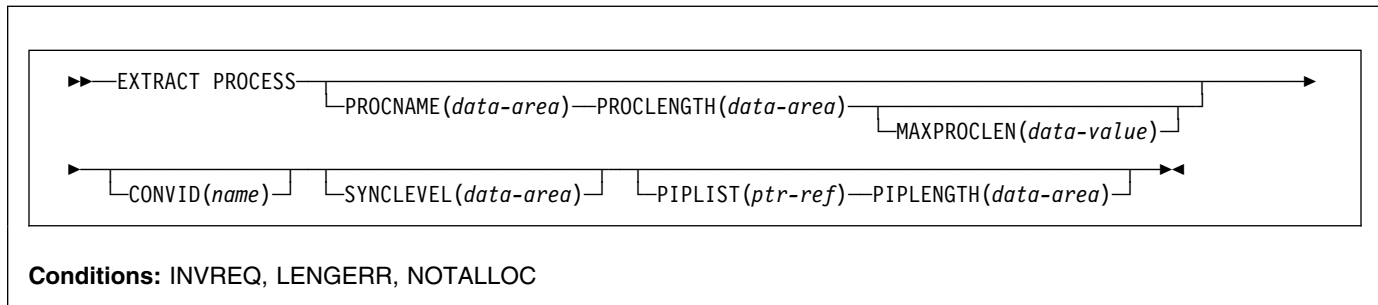
NOTALLOC

occurs if the facility specified in the command is not owned by the application.

Default action: terminate the task abnormally.

EXTRACT PROCESS

Retrieve values from APPC conversation attach header.



Description

EXTRACT PROCESS lets an application program access conversation-related data, specified to CICS when the program is attached. The attach receiver does not have to execute an EXTRACT PROCESS command unless it requires this information.

The EXTRACT PROCESS command is valid only on an APPC conversation that is the principal facility for the task.

Options

CONVID(*name*)

identifies the conversation to which the command relates. The 4-character name identifies the token representing the principal session (EIBTRMID).

For compatibility with earlier releases, SESSION is accepted as a synonym for CONVID. New programs should use CONVID.

If CONVID and SESSION are both omitted, the principal facility for the task is used by default.

MAXPROCLEN(*data-value*)

specifies the buffer length of PROCNAME. If MAXPROCLEN is not specified, the buffer is assumed to have 32 bytes.

PIPELENGTH(*data-area*)

specifies a halfword binary data area in which the total length of the process initialization parameter (PIP) list is returned.

PIPLIST(*ptr-ref*)

specifies a pointer reference that is set to the address of a CICS-provided data area containing a PIP list. This list contains variable-length records in the same format as the list in the CONNECT PROCESS command. A returned value of zero means that no PIP data has been received by CICS.

PROCLENGTH(*data-area*)

specifies a halfword data area that is set by CICS to the length of the process name. If PROCNAME is specified, this option must be specified.

PROCNAME(*data-area*)

specifies the data area to receive the process name specified by the remote system that caused the task to start. The data area can be 1–64 bytes long. The process name is padded on the right with blanks if it is too short. The PROCNAME data area should not be shorter than the MAXPROCLEN value.

SYNCLEVEL(*data-area*)

specifies a halfword data area that is set by CICS to the SYNCLEVEL value. For further information about synchronization levels, see the *CICS Intercommunication Guide*.

Conditions

INVREQ

RESP2 values:

- 200** A distributed program link server application specified the function-shipping session (its principal facility) on the CONVID option.

also occurs (RESP2 not set) in any of the following situations:

- EXTRACT PROCESS has been used on a conversation other than APPC mapped (for example, LUTYPE6.1, APPC basic, or CPI Communications).
- EXTRACT PROCESS has been used on a conversation that was not started by input from the network, and whose session is not a principal facility.
- EXTRACT PROCESS is issued against a CPI-Communications conversation.

Default action: terminate the task abnormally.

LENGERR

occurs if the actual length of PROCNAME is greater than MAXPROCLN, or greater than 32 bytes if MAXPROCLN is not specified.

Default action: terminate the task abnormally.

NOTALLOC

occurs if the specified CONVID value specified does not relate to a conversation owned by the application.

Default action: terminate the task abnormally.

EXTRACT TCT

EXTRACT TCT

Convert an 8-character name to a 4-character name on an LUTYPE6.1 logical unit.

```
▶▶—EXTRACT TCT—NETNAME(name)—▶▶  
      ┌—SYSID(systemname)—┐  
      └—TERMID(data-area)—┘
```

Condition: INVREQ, NOTALLOC

Description

EXTRACT TCT converts the 8-character VTAM network name for a logical unit into the corresponding 4-character name it is known by in the local CICS system.

Options

NETNAME(*name*)

specifies the 8-character name of the logical unit in the VTAM network.

SYSID(*systemname*)

specifies the variable to be set to the equivalent local name of the system.

TERMID(*data-area*)

specifies the variable to be set to the equivalent local name of the terminal.

Conditions

INVREQ

occurs if NETNAME is not valid.

Default action: terminate the task abnormally.

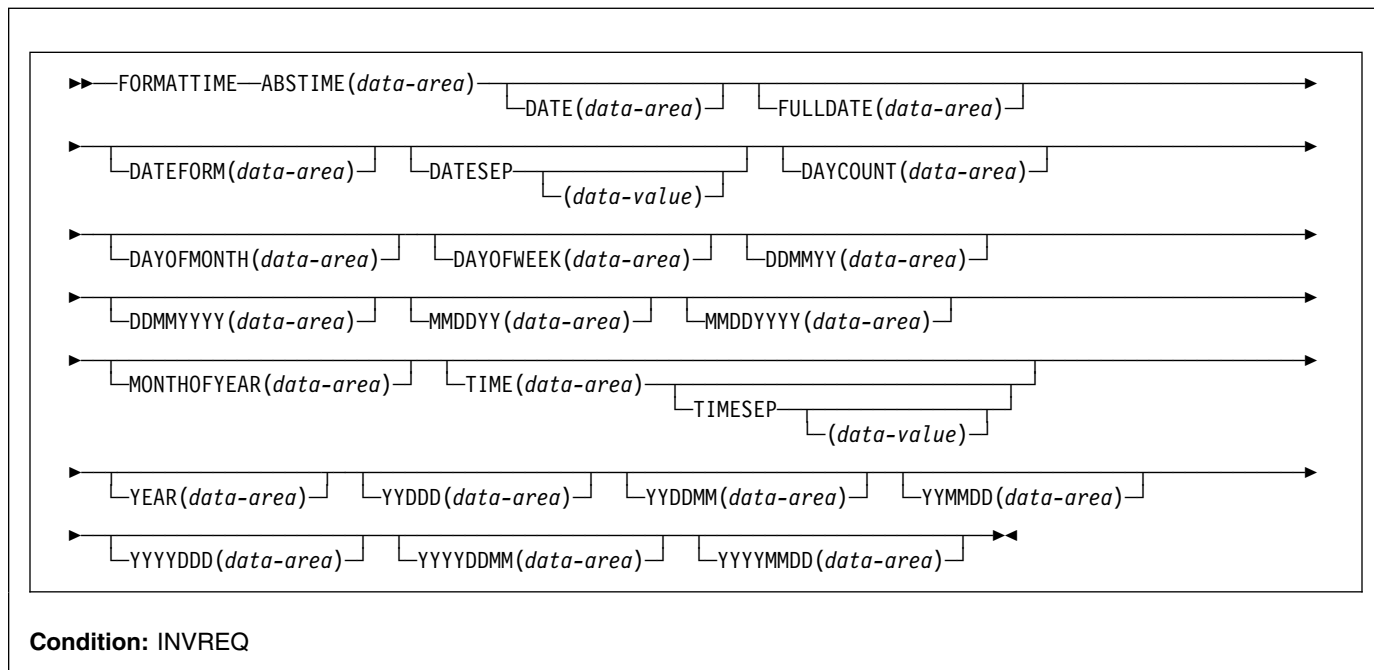
NOTALLOC

occurs if the facility specified in the command is not owned by the application.

Default action: terminate the task abnormally.

FORMATTIME

Transform absolute date and time into a specified format.



Description

FORMATTIME transforms the absolute date and time into any of a variety of formats. Normally, the ABSTIME argument is the value returned by an ASKTIME ABSTIME command.

If the milliseconds come to 500 or more, the returned seconds and, if necessary, the minutes and hours, are rounded up. The day, month, and year are, however, never rounded up. However, in the case where the ABSTIME argument contains a value representing the half-second before midnight, no rounding is performed, and the TIME parameter returns 23:59:59.

To obtain an elapsed time in a particular format, the ABSTIME data value can be the difference between two values returned by ASKTIME, and options such as DAYCOUNT(d) and TIME(t) can be specified.

Options

ABSTIME(*data-area*)

specifies the data value for the time, in packed decimal, since 00:00 hours on 1 January 1900 (in milliseconds rounded to the nearest hundredth of a second) that is to be converted to an alternative format.

The format of the parameter is:

```
COBOL: PIC S9(15) COMP-3
C:      char data_ref[8];
PL/I:   FIXED DEC(15);
ASM:    PL8
```

DATE(*data-area*)

specifies the variable that is to receive the date in the format specified in the DATFORM system initialization parameter. A separator is present if requested by the DATESEP option. If no separator is requested, the returned value is left-justified in the 8-character user field. You should normally use this option only when a date is needed for output purposes. Where a date is needed for analysis, you should request the date in explicit form, for example, using the MMDDYY option.

DATEFORM(*data-area*)

specifies the format of the installation-defined date. CICS returns YYMMDD, DDMMYY, or MMDDYY (six characters) according to the DATFORM system initialization parameter.

DATESEP(*data-value*)

specifies the character to be inserted as the separator between the year and the month, and between the day and the month; or between the year and the day if form YYDDD is specified.

FORMATTIME

If you omit this option, no separator is supplied. If you omit “data-value”, a slash (/) is assumed as the separator.

DAYCOUNT(*data-area*)

returns the number of days since 1 January 1900 (day 1), as a fullword binary number. This is useful if you need to compare the current date with a previous date that has, for example, been stored in a data set.

DAYOFMONTH(*data-area*)

returns the number of the day in the month as a fullword binary number.

DAYOFWEEK(*data-area*)

returns the relative day number of the week as a fullword binary number: Sunday=0, Saturday=6. This number can be converted to a textual form of day in any language.

DDMMYY(*data-area*)

specifies the 8-character user field where CICS is to return the date, in day/month/year format (for example, 21/10/98). A separator is present if requested by the DATESEP option. If no separator is requested, the returned value is left-justified in the 8-character user field.

DDMMYYYY(*data-area*)

specifies the 10-character user field where CICS is to return the date, in day/month/year format (for example 17/06/1995). A separator is present if requested by the DATESEP option. If no separator is requested, the returned value is left-justified in the 10-character user field.

FULLDATE(*data-area*)

specifies the 10-character user field where CICS is to return the date, in the format specified in the DATFORM system initialization parameter, with the year expanded to 4 digits. A separator is present if requested by the DATESEP option. If no separator is requested, the returned value is left-justified in the 10-character user field. You should normally use this option only when a date is needed for output purposes. Where a date is needed for analysis, you should request the date in explicit form, for example, using the MMDDYYYY option.

MMDDYY(*data-area*)

specifies the 8-character user field in which CICS is to return the date, in month/day/year format (for example, 10/21/95). A separator is present if requested by the DATESEP option. If no separator is requested, the returned value is left-justified in the 8-character user field.

MMDDYYYY(*data-area*)

specifies the 10-character user field where CICS is to return the date, in month/day/year format (for example 11/21/1995). A separator is present if requested by the DATESEP option. If no separator is requested, the returned value is left-justified in the 10-character user field.

MONTHOFYEAR(*data-area*)

“data-area” is set to the relative month number of the year as a fullword binary number (January=1, December=12). You can convert this number, in your application program, to the name of the month in any language.

TIME(*data-area*)

“data-area” is set as an 8-character field to the current 24-hour clock time in the form hh:mm:ss, where the separator is specified by the TIMESEP option.

TIMESEP(*data-value*)

specifies the character to be used as the separator in the returned time. If you omit this option, no separator is assumed and six bytes are returned in an 8-character field. If you omit the “data-value”, a colon (:) is used as a separator.

YEAR(*data-area*)

specifies the full 4-figure number of the year as a fullword binary number (for example, 1995, 2001).

YYDDD(*data-area*)

specifies the 6-character user field where CICS is to return the date, in year/day format (for example, 95/301). A separator is present if requested by the DATESEP option. If no separator is requested, the returned value is left-justified in the 6-character user field.

YYDDMM(*data-area*)

specifies the 8-character user field where CICS is to return the date, in year/day/month format (for example, 95/30/10). A separator is present if requested by the DATESEP option. If no separator is requested, the returned value is left-justified in the 8-character user field.

YYMMDD(*data-area*)

specifies the 8-character user field where CICS is to return the date, in year/month/day format (for example, 95/10/21). A separator is present if requested by the DATESEP option. If no separator is requested, the returned value is left-justified in the 8-character user field.

YYYYDDD(*data-area*)

specifies the 8-character user field where CICS is to return the date, in year/day format (for example 1995/200). A separator is present if requested by the DATESEP option. If no separator is requested, the returned value is left-justified in the 8-character user field.

YYYYDDMM(*data-area*)

specifies the 10-character user field where CICS is to return the date, in year/day/month format (for example 1995/21/06). A separator is present if requested by the DATESEP option. If no separator is requested, the returned value is left-justified in the 10-character user field.

YYYYMMDD(*data-area*)

specifies the 10-character user field where CICS is to return the date, in year/month/day format (for example 1995/06/21). A separator is present if requested by the DATESEP option. If no separator is requested, the returned value is left-justified in the 10-character user field.

Conditions**INVREQ**

RESP2 values:

- 1 The ABSTIME option is in an incorrect form.

Default action: terminate the task abnormally.

Examples

The following example shows the effect of some of the options of the command. Let “utime” contain the value 002837962864828 in milliseconds.

```
EXEC CICS ASKTIME ABSTIME(utime)
EXEC CICS FORMATTIME ABSTIME(utime)
          DATESEP('-') DDMMYY(date)
          TIME(time) TIMESEP
```

This gives the values 06-12-89 for “date” and 19:01:05 for “time”.

FREE

FREE

Return a terminal or logical unit.

▶▶—FREE—◀◀

Condition: NOTALLOC

Description

FREE returns a terminal or logical unit when the transaction owning it no longer requires it. The principal facility is freed.

If you are running CEDF, and the transaction frees the principal facility, CEDF is terminated.

Conditions

NOTALLOC

occurs if the task is not associated with the terminal.

Default action: terminate the task abnormally.

FREE (APPC)

Return an APPC mapped session to CICS.

```

▶▶ FREE [CONVID(name)] [STATE(cvda)] ◀◀

```

Conditions: INVREQ, NOTALLOC

Description

FREE returns an APPC session to CICS when a transaction owning it no longer requires it. The session can then be allocated for use by other transactions.

If you omit CONVID, the principal facility is freed. Facilities not freed explicitly are freed by CICS when the task terminates.

If you are running CEDF, and the transaction frees the principal facility, CEDF is terminated.

Options

CONVID(*name*)

identifies the APPC mapped session to be freed. The 4-character name identifies either the token returned by a previously executed ALLOCATE command in EIBRSRCE in the EIB, or the token representing the principal session (returned by a previously executed ASSIGN command).

If this option is omitted, the principal facility is assumed.

STATE(*cvda*)

gets the state of the current conversation. The STATE option on a FREE command returns a cvda code of 00 if there is no longer an active conversation. The other output cvda values are:

```

ALLOCATED
CONFFREE
CONFRECEIVE
CONFSEND
FREE
PENDFREE
PENDRECEIVE
RECEIVE
ROLLBACK
SEND
SYNCFREE
SYNCRECEIVE
SYNCSEND

```

Conditions

INVREQ

RESP2 values:

- 200** A distributed program link server application specified the function-shipping session (its principal facility) on the CONVID option.

also occurs (RESP2 not set) in any of the following situations:

- The CONVID value specified in the command relates to a basic (unmapped) APPC conversation.
- The CONVID value specified in the command relates to a CPI-Communications conversation.

Default action: terminate the task abnormally.

NOTALLOC

occurs if the specified CONVID value does not relate to a conversation owned by the application.

Default action: terminate the task abnormally.

FREE (LUTYPE6.1)

FREE (LUTYPE6.1)

Return LUTYPE6.1 sessions to CICS.



Description

FREE returns an LUTYPE6.1 session to CICS when a transaction owning it no longer requires it. The session can then be allocated for use by other transactions.

If you omit both CONVID and SESSION, the principal facility is freed. Facilities not freed explicitly are freed by CICS when the task terminates.

If you are running CEDF, and the transaction frees the principal facility, CEDF is terminated.

Options

CONVID(*name*)

identifies the LUTYPE6.1 session to be freed. The 4-character name identifies either the token returned by a previously executed ALLOCATE command in EIBRSRCE in the EIB, or the token representing the principal session (returned by a previously executed ASSIGN command).

SESSION(*name*)

specifies the symbolic identifier (1–4 characters) of a session TCTTE. This option specifies the alternate facility to be used.

Conditions

INVREQ

occurs if the session specified in the command was allocated for a basic (unmapped) APPC conversation.

(See also EIBRCODE in Appendix A, “EXEC interface block” on page 373.)

Default action: terminate the task abnormally.

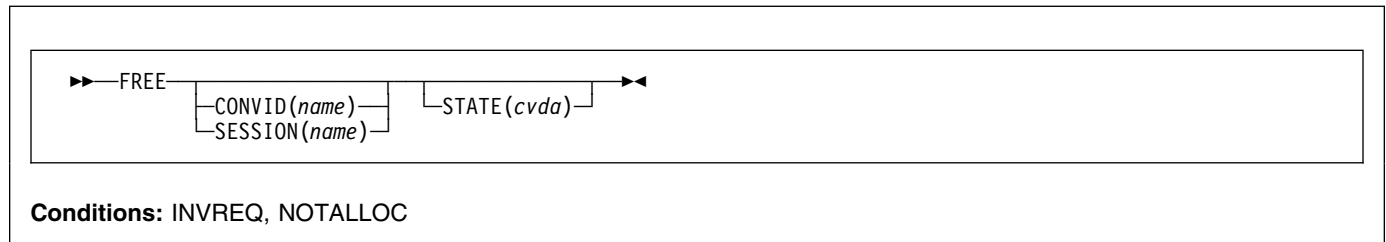
NOTALLOC

occurs if the session specified in the command is not owned by the application.

Default action: terminate the task abnormally.

FREE (MRO)

Return MRO sessions to CICS.



Description

FREE returns an MRO session to CICS when a transaction owning it no longer requires it. The session can then be allocated for use by other transactions.

If you omit both CONVID and SESSION, the principal facility is freed. Facilities not freed explicitly are freed by CICS when the task terminates.

If you are running CEDF, and the transaction frees the principal facility, CEDF is terminated.

Options

CONVID(*name*)

identifies the MRO session to be freed. The 4-character name identifies either the token returned by a previously executed ALLOCATE command in EIBRSRCE in the EIB, or the token representing the principal session (returned by a previously executed ASSIGN command).

SESSION(*name*)

specifies the symbolic identifier (1–4 characters) of a session TCTTE. This option specifies the alternate facility to be used.

STATE(*cvda*)

gets the state of the current conversation. The STATE on a FREE command returns a *cvda* code of 00 if there is no longer an active conversation. The other output *cvda* values are:

```

ALLOCATED
FREE
PENDFREE
RECEIVE
ROLLBACK
SEND
SYNCFREE
SYNCRECEIVE
SYNCSEND

```

Conditions

INVREQ

occurs in any one of the following situations:

- The session specified in the command was allocated for a basic (unmapped) APPC conversation
- The session is in the wrong state to be freed.

(See also EIBRCODE in Appendix A, “EXEC interface block” on page 373.)

Default action: terminate the task abnormally.

NOTALLOC

occurs if the session specified in the command is not owned by the application.

Default action: terminate the task abnormally.

FREEMAIN

Release main storage acquired by a GETMAIN command.

```

  ►► FREEMAIN — DATA(data-area) —►►
                └─ DATAPOINTER(ptr-value) ┘

```

Condition: INVREQ

Note for dynamic transaction routing

Using FREEMAIN of storage GETMAINED with SHARED, or of a PROGRAM resource defined with the RDO option RELOAD(YES), that has been LOADED, could create inter-transaction affinities that adversely affect the use of dynamic transaction routing. See the *CICS Application Programming Guide* for more information about transaction affinities.

Description

FREEMAIN releases main storage previously acquired by a GETMAIN command issued by the application, or by a LOAD for a program with RELOAD(YES) specified on its RDO PROGRAM resource definition. If the task that GETMAINED the storage or LOADED the program does not release it, CICS releases it at task end, unless:

- The GETMAIN command specified the SHARED option.
- The program is defined with RELOAD(YES).
- The program is defined with RELOAD(NO) but was LOADED with the HOLD option.

In the first two cases, the storage remains allocated until some other task issues a FREEMAIN to release it. In the third case, the program remains available until it is RELEASED by another task.

You can release CICS-key storage from a program only if it is being executed in CICS key. If the previously-acquired storage was obtained from CICS-key storage, and the program issuing the FREEMAIN is in user-key, an INVREQ condition occurs with a RESP2 value of 2.

Options

DATA(*data-area*)

specifies the data area of main storage to be released.

This storage must have been acquired by a previous GETMAIN command, except in the case of BMS pages. (For more guidance about BMS pages, see the description of the SET option in the *CICS Application Programming Guide*.)

Note that this option specifies the data area that was acquired by the GETMAIN command, not the pointer reference that was set to that address. You must use

the DATAPOINTER option to specify a pointer-reference: DATA and DATAPOINTER are mutually exclusive. Therefore, in assembler language, “data-area” must be a relocatable expression that is a data reference; in COBOL or C it must be a data name; and in PL/I it must be a data reference. (See the *CICS Application Programming Guide* for a discussion of argument values.)

The length of storage released is the length obtained by the GETMAIN and not necessarily the length of the data area.

DATAPOINTER(*ptr-value*)

specifies the address of the main storage to be released. This option is an alternative to the DATA option, and specifies the pointer reference that was returned by a GETMAIN command using the SET option.

The length of storage released is the length obtained by the GETMAIN.

Conditions

INVREQ

RESP2 values:

- 1 The storage specified by the DATA or DATAPOINTER parameter is not storage acquired by a GETMAIN command.
- 2 The storage area specified by the DATA or DATAPOINTER parameter is in CICS-key storage, and the program issuing the FREEMAIN command is in user-key.

Default action: terminate the task abnormally.

Examples

Here are some examples of the use of FREEMAIN.

COBOL:

```
DATA DIVISION.
WORKING-STORAGE SECTION.
77 AREA-POINTER    USAGE IS POINTER.
LINKAGE SECTION.
  01 WORKAREA      PIC X(100).
PROCEDURE DIVISION.
  EXEC CICS GETMAIN SET(AREA-POINTER)
  LENGTH(100)
  END-EXEC.
  .
  SET ADDRESS OF WORKAREA TO AREA-POINTER.
  .
  .
  EXEC CICS FREEMAIN DATA(WORKAREA)
  END-EXEC.
  EXEC CICS RETURN
  END-EXEC.
```

Alternatively, the previous COBOL example could free the storage by the following command:

```
EXEC CICS FREEMAIN DATAPOINTER(AREA-POINTER)
END-EXEC.
```

C:

```
#pragma XOPTS(CICS);
#define MAINSIZE 100;
main()
{
  char          *buffer;
  struct eib_record dfheiptr;
  EXEC CICS ADDRESS EIB(dfheiptr);
  EXEC CICS GETMAIN SET(buffer)
  LENGTH(MAINSIZE);
  buffer[2] = 'a';
  .
  .
  EXEC CICS FREEMAIN DATA(buffer);
  EXEC CICS RETURN;
}
```

PL/I:

```
DCL AREA_PTR    POINTER,
WORKAREA      CHAR(100) BASED(AREA_PTR);
.
.
.
EXEC CICS GETMAIN SET(AREA_PTR) LENGTH(100);
.
EXEC CICS FREEMAIN DATA(WORKAREA);
```

Assembler:

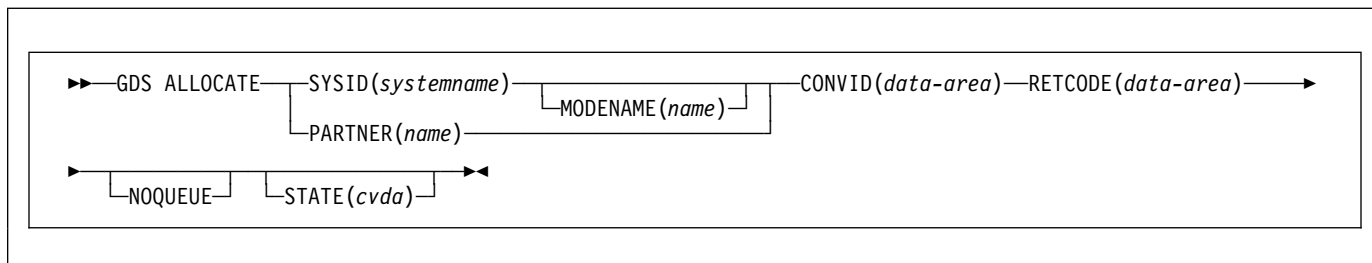
```
WORKAREA DS CL100
.
.
EXEC CICS GETMAIN SET(9) LENGTH(100)
USING WORKAREA,9
EXEC CICS FREEMAIN DATA(WORKAREA)
```

Alternatively, you can free storage using the DATAPOINTER option as shown in the following example:

```
WORKAREA DS CL100
.
EXEC CICS GETMAIN SET(9) LENGTH(100)
USING WORKAREA,9
.
.
DROP 9
.
EXEC CICS FREEMAIN DATAPOINTER(9)
```

GDS ALLOCATE

Acquire a session to a remote system for use by APPC basic conversation (assembler-language and C programs only).



Description

GDS ALLOCATE acquires a session to a remote system.

The return code is given in RETCODE (see Table 3 on page 105). For a list of return code values, see the *CICS Distributed Transaction Programming Guide*. EXEC CICS conditions are never raised on GDS commands.

Options

In general, the arguments can be replaced by absolute or relocatable assembler-language expressions.

CONVID(*data-area*)

specifies the 4-character application data area that is to contain the token returned by an ALLOCATE command to identify the allocated conversation. This token is required in subsequent GDS commands issued on the conversation.

MODENAME(*name*)

specifies the name of the mode group from which the session is to be acquired. If you specify SYSID and omit MODENAME, CICS selects a modename from those defined for the system.

NOQUEUE

specifies that the request to allocate a session is not to be queued when a suitable APPC session cannot be acquired immediately. A session is acquired immediately only if it is a bound contention winner that is not already allocated to another conversation.

The return code in RETCODE indicates whether or not a session has been acquired.

If the NOQUEUE option is not used, a delay may occur before control is passed back to the application program. A delay can occur for any of the following reasons:

- All sessions for the specified SYSID and MODENAME are in use.

- The CICS allocation algorithm has selected a session that is not currently bound (in which case, CICS has to bind).
- The CICS allocation algorithm has selected a contention loser (in which case, CICS has to bid).

If there is a delay, the program waits until the session has been acquired.

PARTNER(*name*)

specifies the name (eight characters) of a set of definitions that include the names of a remote LU (NETNAME) and a communication profile to be used on the allocated session. For APPC basic conversations, the only relevant attribute set by the profile is MODENAME.

If you use this option as an alternative to SYSID and MODENAME, CICS uses the NETNAME and MODENAME from the PARTNER definition.

RETCODE(*data-area*)

specifies the 6-byte application data area into which return code information is to be moved.

STATE(*cvda*)

gets the state of the current conversation. The *cvda* values returned by CICS are:

```

  ALLOCATED
  CONFFREE
  CONFRECEIVE
  CONFSEND
  FREE
  PENDFREE
  PENDRECEIVE
  RECEIVE
  ROLLBACK
  SEND
  SYNCFREE
  SYNCRECEIVE
  SYNCSEND
  
```

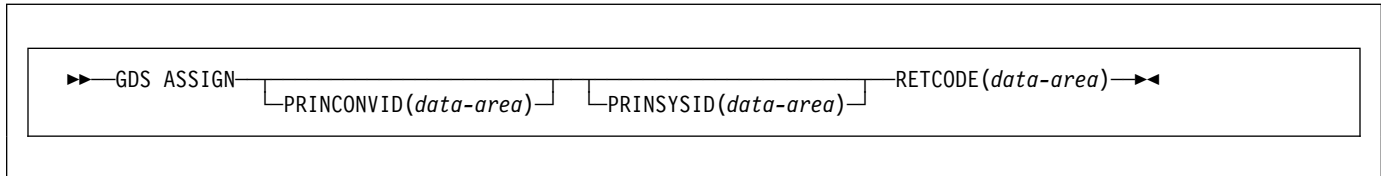
SYSID(*systemname*)

specifies the remote system to which an APPC session is to be allocated. The name, which is 1–4 characters, identifies an entry (defined as an APPC connection) in the CICS terminal control table.

RETCODE (hexadecimal)	Description
01 0C 00	SYSID is unrecognized.
01 0C 04	SYSID is not an LUTYPE6.2 connection name.
01 04 04	NOQUEUE is specified but no bound connection-winner sessions are available.
01 04 08	MODENAME is not known.
01 04 0C	The MODENAME value is SNASVCMG which is restricted to use by CICS.
01 04 0C	VTAM has no class of service (COS) table for the MODENAME value.
01 04 10	The task was canceled during queuing of the command.
01 04 14	All modegroups are closed.
01 04 14	The requested modegroup is closed.
01 04 18	The requested modegroup is draining (closing down).
01 08 00	All sessions in the requested modegroup are unusable.
01 08 00	The connection is in quiesce state.
01 08 00	The connection is out of service.
01 08 00	The connection is not acquired.
01 08 00	The requested modegroup's local max (maximum permitted number of sessions) is 0.
01 08 00	The VTAM ACB is closed.
01 0C 14	The NETNAME specified in the PARTNER definition is not known.
02 0C 00	PARTNER is not known.
06 00 00	The PROFILE specified in the PARTNER definition is not known.

GDS ASSIGN

Get identifier of principal facility in use by APPC basic conversation (assembler-language and C programs only).



Description

GDS ASSIGN gets the identifier of the principal facility.

The return code is given in RETCODE (see Table 4). For a list of return code values, see the *CICS Distributed Transaction Programming Guide*. EXEC CICS conditions are never raised on GDS commands.

Options

In general, the arguments can be replaced by absolute or relocatable assembler-language expressions.

PRINCONVID(data-area)

specifies a 4-byte data area in which the conversation token (CONVID) of the principal facility is to be returned.

PRINSYSID(data-area)

specifies a 4-byte data area in which the SYSID of the principal facility is to be returned.

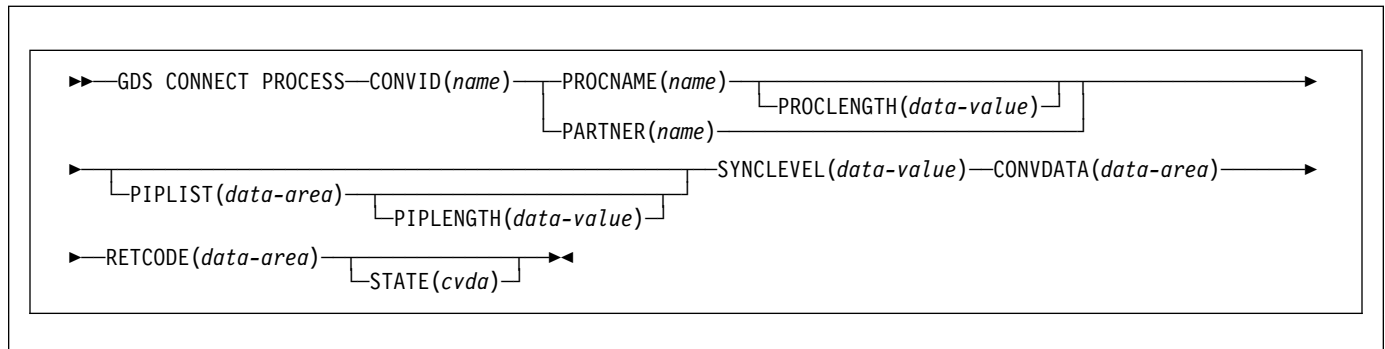
RETCODE(data-area)

specifies the 6-byte application data area into which return code information is to be moved.

Table 4. GDS ASSIGN return codes	
RETCODE (hexadecimal)	Description
03 00	Principal facility is not APPC.
03 04	Principal facility is not basic.
04	No terminal principal facility exists.

GDS CONNECT PROCESS

Initiate APPC basic conversation (assembler-language and C programs only).



Description

EXEC CICS conditions are never raised on GDS commands.

The return code is given in RETCODE (see Table 5 on page 108). For a list of return code values, see the *CICS Distributed Transaction Programming Guide*. EXEC CICS conditions are never raised on GDS commands.

GDS CONNECT PROCESS allows the application program to specify a partner application that is to run in the remote system.

Options

In general, the arguments can be replaced by absolute or relocatable assembler-language expressions.

CONVDATA(data-area)

specifies the 24-byte application data area into which conversation-related information is to be moved. A description of the format of the data area is given in the discussion of CONVDATA fields in the *CICS Distributed Transaction Programming Guide*.

CONVID(name)

identifies the conversation to which the command relates. The 4-character name identifies either the token returned by a previously executed GDS ALLOCATE command, or the token representing the principal session (returned by a previously executed GDS ASSIGN command).

PARTNER(name)

specifies the name (8 characters) of a set of definitions that includes the name (or extended name) of a remote partner transaction (TPNAME or XTPNAME). You can use this option as an alternative to PROCNAME and PROCLENGTH.

PIPELENGTH(data-value)

specifies the total length of the process initialization parameter (PIP) list specified on a CONNECT PROCESS command.

PIPELIST(data-area)

specifies the PIP data that is to be sent to the remote process. See the *CICS Intercommunication Guide* for information about PIP data.

PROCLENGTH(data-value)

specifies the length (as a halfword binary value in the range 1–64) of the target process name.

PROCNAME(name)

specifies the name of the remote application. The APPC architecture allows names of lengths (1–64 bytes), but leaves each product free to set its own maximum. If the remote system is CICS, you can use the standard 4-character transaction ID.

RETCODE(data-area)

specifies the 6-byte application data area into which return code information is to be moved.

STATE(cvda)

gets the state of the current conversation. The cvda values returned by CICS are:

```

ALLOCATED
CONFFREE
CONFRECEIVE
CONFSEND
FREE
PENDFREE
PENDRECEIVE
RECEIVE
ROLLBACK
SEND
SYNCFREE
SYNCRECEIVE
SYNCSSEND
  
```

GDS CONNECT PROCESS

SYNCLEVEL(*data-value*)

specifies the synchronization level (halfword binary value) desired for the current conversation. The possible values are:

- 0 None
- 1 Confirm
- 2 Syncpoint

RETCODE (hexadecimal)	Description
02 0C 00	PARTNER is not known.
03 00	CONVID is for a session that is not APPC.
03 00	CONVID is for a session that is in use by CPI Communications.
03 04	CONVID is for a conversation that is not basic.
03 0C	The SYNCLEVEL option specifies a value other than 0, 1, or 2.
03 0C	The SYNCLEVEL option requested either 1 or 2, but it was unavailable.
03 08	A state check occurred.
04	CONVID is for a session that is not allocated to the task, or that is a relay link.
05 00 00 00 00 20	PROCLENGTH is outside the range 1–64.
05 00 00 00 7F FF	The PIPELENGTH value is outside the range 4–763.
05 00 00 00 7F FF	The 2-byte length field (LL) for one of the PIPs is less than 4.
05 00 00 00 7F FF	The total of the LLs in PIP data is greater than the PIPELENGTH value.

GDS EXTRACT ATTRIBUTES

Access state information on an APPC basic conversation (assembler-language and C programs only).

```

▶▶ GDS EXTRACT ATTRIBUTES—CONVID(name)—STATE(cvda)—CONVDATA(data-area)—RETCODE(data-area)—▶▶

```

Description

GDS EXTRACT ATTRIBUTES accesses state information about an APPC basic conversation.

The return code is given in RETCODE (see Table 6). For a list of return code values, see the *CICS Distributed Transaction Programming Guide*. EXEC CICS conditions are never raised on GDS commands.

Options

In general, the arguments can be replaced by absolute or relocatable assembler-language expressions.

CONVID(*name*)

identifies the conversation to which the command relates. The 4-character name identifies either the token returned by a previously executed GDS ALLOCATE command, or the token representing the principal session (returned by a previously executed GDS ASSIGN command).

CONVDATA(*data-area*)

specifies the 24-byte application data area into which conversation-related information is to be moved. A description of the format of the data area is given in the discussion of CONVDATA fields in the *CICS Distributed Transaction Programming Guide*.

RETCODE(*data-area*)

specifies the 6-byte application data area into which return code information is to be moved.

STATE(*cvda*)

gets the state of the current conversation. The cvda values returned by CICS are:

```

ALLOCATED
CONFFREE
CONFRECEIVE
CONFSEND
FREE
PENDFREE
PENDRECEIVE
RECEIVE
ROLLBACK

```

```

SEND
SYNCFREE
SYNCRECEIVE
SYNCSSEND

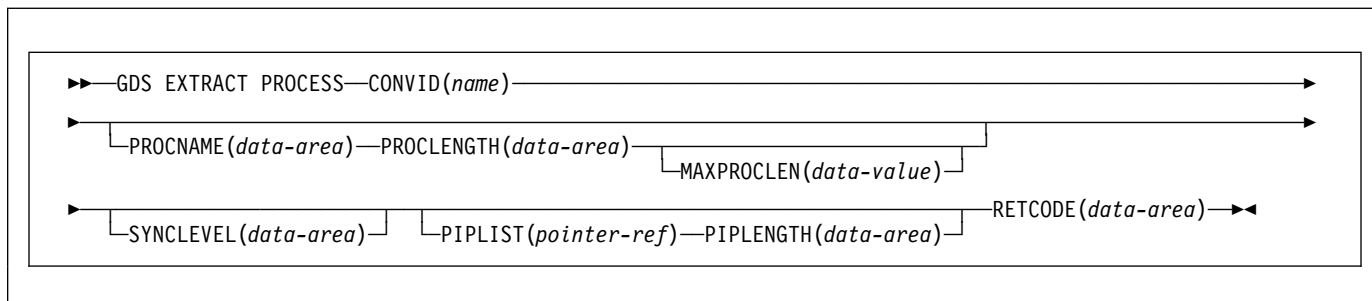
```

Table 6. GDS EXTRACT ATTRIBUTES return codes

RETCODE (hexadecimal)	Description
03 00	CONVID is for a session that is not APPC.
03 00	CONVID is for a session that is in use by CPI Communications.
03 01	INVREQ for a DPL server program.
03 04	CONVID is for a conversation that is not basic.
04	CONVID is for a session that is not allocated to the task, or that is a relay link.

GDS EXTRACT PROCESS

Retrieve values from an APPC basic conversation (assembler-language and C programs only).



Description

GDS EXTRACT PROCESS retrieves values from an APPC basic conversation. The data retrieved is valid only when the command is issued against an APPC basic principal facility.

The return code is given in RETCODE (see Table 7). For a list of return code values, see the *CICS Distributed Transaction Programming Guide*. EXEC CICS conditions are never raised on GDS commands.

Options

In general, the arguments can be replaced by absolute or relocatable assembler-language expressions.

CONVID(name)

identifies the conversation the command relates to. The 4-character name identifies the token representing the principal session (returned by a previously executed GDS ASSIGN command).

MAXPROCLEN(data-value)

specifies the length (1–64 characters) of the PROCNAME data area. If MAXPROCLEN is not specified, the buffer is assumed to have 32 bytes.

PIPLENGTH(data-area)

specifies a halfword binary data area that is to receive the length of the PIPLIST received by a GDS EXTRACT PROCESS command.

PIPLIST(pointer-ref)

specifies the pointer reference that is to be set to the address of the PIPLIST received by a GDS EXTRACT PROCESS command. A zero setting indicates that no PIPLIST was received.

PROCLENGTH(data-area)

specifies a halfword binary data area that is set to the actual length of the process name.

PROCNAME(data-area)

specifies the application target data area (1–64 bytes) into which the process name, specified in the APPC attach function management header, is to be moved. The area is padded with blanks, if necessary.

RETCODE(data-area)

specifies the 6-byte application data area into which return code information is to be moved.

SYNCLEVEL(data-area)

specifies a halfword binary data area that is set to indicate the synchronization level in effect for the current conversation. The possible values are:

- 0 None
- 1 Confirm
- 2 Syncpoint

Table 7. GDS EXTRACT PROCESS return codes

RETCODE (hexadecimal)	Description
03 00	CONVID is for a session that is not APPC.
03 00	CONVID is for a session that is in use by CPI Communications.
03 00	CONVID is for a session that is not the principal facility.
03 00	Principal facility was not started by terminal data.
03 04	CONVID is for a conversation that is not basic.
04	CONVID is for a session that is not allocated to the task, or that is a relay link.
05 00 00 00 00 20	PROCLENGTH value returned is greater than MAXPROCLEN value.

GDS FREE

Return an APPC session to CICS (assembler-language and C programs only).

```

▶▶ GDS FREE—CONVID(name)—CONVDATA(data-area)—RETCODE(data-area)—STATE(cvda)—▶▶

```

Description

GDS FREE returns the session to CICS. The issue of this command is valid only when the conversation is finished, that is, the conversation state is FREE.

The return code is given in RETCODE (see Table 8). For a list of return code values, see the *CICS Distributed Transaction Programming Guide*. EXEC CICS conditions are never raised on GDS commands.

Options

In general, the arguments can be replaced by absolute or relocatable assembler-language expressions.

CONVDATA(*data-area*)

specifies the 24-byte application data area into which conversation-related information is to be moved. A description of the format of the data area is given in the discussion of CONVDATA fields in the *CICS Distributed Transaction Programming Guide*.

CONVID(*name*)

identifies the conversation to be freed. The 4-character name identifies either the token returned by a previously executed GDS ALLOCATE command, or the token representing the principal session (returned by a previously executed GDS ASSIGN command).

RETCODE(*data-area*)

specifies the 6-byte application data area into which return code information is to be moved.

STATE(*cvda*)

gets the state of the current conversation. The STATE on a FREE command returns a cvda code of 00 if there is no longer an active conversation. The other output cvda values are:

```

ALLOCATED
CONFFREE
CONFRECEIVE
CONFSEND
FREE
PENDFREE
PENDRECEIVE

```

```

RECEIVE
ROLLBACK
SEND
SYNCFREE
SYNCRECEIVE
SYNCSSEND

```

Table 8. GDS FREE return codes

RETCODE (hexadecimal)	Description
03 00	CONVID is for a session that is not APPC.
03 00	CONVID is for a session that is in use by CPI Communications.
03 04	CONVID is for a conversation that is not basic.
03 08	A state check has occurred.
04	CONVID is for a session that is not allocated to the task, or that is a relay link.

GDS ISSUE ABEND

Terminate APPC basic conversation abnormally (assembler-language and C programs only).

```

▶▶ GDS ISSUE ABEND—CONVID(name)—CONVDATA(data-area)—RETCODE(data-area)—STATE(cvda)◀◀

```

Description

GDS ISSUE ABEND causes an APPC basic conversation to end immediately, regardless of the conversation state. The partner transaction is informed.

The return code is given in RETCODE (see Table 9). For a list of return code values, see the *CICS Distributed Transaction Programming Guide*. EXEC CICS conditions are never raised on GDS commands.

Options

In general, the arguments can be replaced by absolute or relocatable assembler-language expressions.

CONVDATA(*data-area*)

specifies the 24-byte application data area into which conversation-related information is to be moved. A description of the format of the data area is given in the discussion of CONVDATA fields in the *CICS Distributed Transaction Programming Guide*.

CONVID(*name*)

identifies the conversation to which the command relates. The 4-character name identifies either the token returned by a previously executed GDS ALLOCATE command, or the token representing the principal session (returned by a previously executed GDS ASSIGN command).

RETCODE(*data-area*)

specifies the 6-byte application data area into which return code information is to be moved.

STATE(*cvda*)

gets the state of the current conversation. The *cvda* values returned by CICS are:

```

ALLOCATED
CONFFREE
CONFRECEIVE
CONFSEND
FREE
PENDFREE
PENDRECEIVE
RECEIVE

```

```

ROLLBACK
SEND
SYNCFREE
SYNCRECEIVE
SYNCSEND

```

Table 9. GDS ISSUE ABEND return codes

RETCODE (hexadecimal)	Description
03 00	CONVID is for a session that is not APPC.
03 00	CONVID is for a session that is in use by CPI Communications.
03 04	CONVID is for a conversation that is not basic.
03 08	A state check has occurred.
04	CONVID is for a session that is not allocated to the task, or that is a relay link.

GDS ISSUE CONFIRMATION

Issue synchronization request on APPC basic conversation (assembler-language and C programs only).

```

▶▶ GDS ISSUE CONFIRMATION—CONVID(name)—CONVDATA(data-area)—RETCODE(data-area)—STATE(cvda)▶▶

```

Description

GDS ISSUE CONFIRMATION issues a synchronization request in response to a GDS SEND CONFIRM issued by a partner transaction.

The return code is given in RETCODE (see Table 10). For a list of return code values, see the *CICS Distributed Transaction Programming Guide*. EXEC CICS conditions are never raised on GDS commands.

Options

In general, the arguments can be replaced by absolute or relocatable assembler-language expressions.

CONVDATA(*data-area*)

specifies the 24-byte application data area into which conversation-related information is to be moved. A description of the format of the data area is given in the discussion of CONVDATA fields in the *CICS Distributed Transaction Programming Guide*.

CONVID(*name*)

identifies the conversation to which the command relates. The 4-character name identifies either the token returned by a previously executed GDS ALLOCATE command, or the token representing the principal session (returned by a previously executed GDS ASSIGN command).

RETCODE(*data-area*)

specifies the 6-byte application data area into which return code information is to be moved.

STATE(*cvda*)

gets the state of the current conversation. The *cvda* values returned by CICS are:

```

ALLOCATED
CONFFREE
CONFRECEIVE
CONFSEND
FREE
PENDFREE
PENDRECEIVE
RECEIVE

```

```

ROLLBACK
SEND
SYNCFREE
SYNCRECEIVE
SYNCSEND

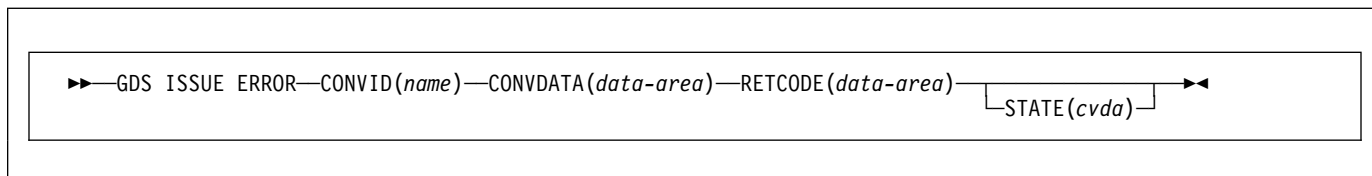
```

Table 10. GDS ISSUE CONFIRMATION return codes

RETCODE (hexadecimal)	Description
03 00	CONVID is for a session that is not APPC.
03 00	CONVID is for a session that is in use by CPI Communications.
03 04	CONVID is for a conversation that is not basic.
03 08	A state check has occurred.
03 14	The command was issued for a sync level 0 conversation.
04	CONVID is for a session that is not allocated to the task, or that is a relay link.

GDS ISSUE ERROR

Inform APPC basic conversation partner of error (assembler-language and C programs only).



Description

GDS ISSUE ERROR informs the conversation partner that there is an error.

The return code is given in RETCODE, see below. For a list of return code values, see the *CICS Distributed Transaction Programming Guide*. EXEC CICS conditions are never raised on GDS commands.

Options

In general, the arguments can be replaced by absolute or relocatable assembler-language expressions.

CONVDATA(*data-area*)

specifies the 24-byte application data area into which conversation-related information is to be moved. A description of the format of the data area is given in the discussion of CONVDATA fields in the *CICS Distributed Transaction Programming Guide*.

CONVID(*name*)

identifies the conversation to which the command relates. The 4-character name identifies either the token returned by a previously executed GDS ALLOCATE command, or the token representing the principal session (returned by a previously executed GDS ASSIGN command).

RETCODE(*data-area*)

specifies the 6-byte application data area into which return code information is to be moved.

STATE(*cvda*)

gets the state of the current conversation. The *cvda* values returned by CICS are:

```

ALLOCATED
CONFFREE
CONFRECEIVE
CONFSEND
FREE
PENDFREE
PENDRECEIVE
RECEIVE
ROLLBACK

```

```

SEND
SYNCFREE
SYNCRECEIVE
SYNCSSEND

```

Table 11. GDS ISSUE ERROR return codes

RETCODE (hexadecimal)	Description
03 00	CONVID is for a session that is not APPC.
03 00	CONVID is for a session that is in use by CPI Communications.
03 04	CONVID is for a conversation that is not basic.
03 08	A state check has occurred.
04	CONVID is for a session that is not allocated to task, or that is a relay link.

GDS ISSUE PREPARE

Issue first flow of syncpoint request on APPC basic conversation (assembler-language and C programs only).

```

▶▶ GDS ISSUE PREPARE—CONVID(name)—CONVDATA(data-area)—RETCODE(data-area)—STATE(cvda)—◀◀

```

Description

GDS ISSUE PREPARE issues first flow of syncpoint request.

The return code is given in RETCODE (see Table 12). For a list of return code values, see the *CICS Distributed Transaction Programming Guide*. EXEC CICS conditions are never raised on GDS commands.

Options

In general, the arguments can be replaced by absolute or relocatable assembler-language expressions.

CONVDATA(*data-area*)

specifies the 24-byte application data area into which conversation-related information is to be moved. A description of the format of the data area is given in the discussion of CONVDATA fields in the *CICS Distributed Transaction Programming Guide*.

CONVID(*name*)

identifies the conversation to which the command relates. The 4-character name identifies either the token returned by a previously executed GDS ALLOCATE command, or the token representing the principal session (returned by a previously executed GDS ASSIGN command).

RETCODE(*data-area*)

specifies the 6-byte application data area into which return code information is to be moved.

STATE(*cvda*)

gets the state of the current conversation. The *cvda* values returned by CICS are:

```

ALLOCATED
CONFFREE
CONFRECEIVE
CONFSEND
FREE
PENDFREE
PENDRECEIVE
RECEIVE
ROLLBACK
SEND

```

```

SYNCFREE
SYNCRECEIVE
SYNCSEND

```

Table 12. GDS ISSUE PREPARE return codes

RETCODE (hexadecimal)	Description
03 00	CONVID is for a session that is not APPC.
03 00	CONVID is for a session that is in use by CPI Communications.
03 04	CONVID is for a conversation that is not basic.
03 0C	The command was issued on a conversation that is not sync-level 2.
03 24	A state error occurred.
04	CONVID is for a session that is not allocated to task, or that is a relay link.

GDS ISSUE SIGNAL

Request change of direction from sending transaction APPC basic conversation (assembler-language and C programs only).

```

▶▶—GDS ISSUE SIGNAL—CONVID(name)—CONVDATA(data-area)—RETCODE(data-area)—STATE(cvda)—◀◀

```

Description

GDS ISSUE SIGNAL requests a change of direction.

The return code is given in RETCODE (see Table 13). For a list of return code values, see the *CICS Distributed Transaction Programming Guide*. EXEC CICS conditions are never raised on GDS commands.

Options

In general, the arguments can be replaced by absolute or relocatable assembler-language expressions.

CONVDATA(*data-area*)

specifies the 24-byte application data area into which conversation-related information is to be moved. A description of the format of the data area is given in the discussion of CONVDATA fields in the *CICS Distributed Transaction Programming Guide*.

CONVID(*name*)

identifies the conversation to which the command relates. The 4-character name identifies either the token returned by a previously executed GDS ALLOCATE command, or the token representing the principal session (returned by a previously executed GDS ASSIGN command).

RETCODE(*data-area*)

specifies the 6-byte application data area into which return code information is to be moved.

STATE(*cvda*)

gets the state of the current conversation. The *cvda* values returned by CICS are:

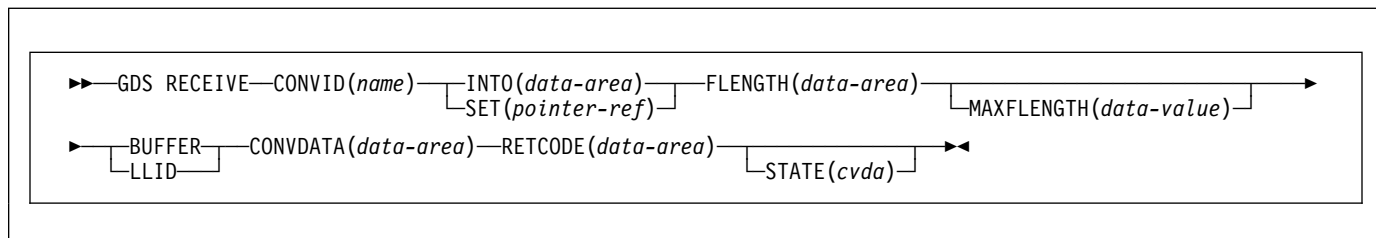
ALLOCATED
 CONFFREE
 CONFRECEIVE
 CONFSEND
 FREE
 PENDFREE
 PENDRECEIVE
 RECEIVE
 ROLLBACK
 SEND
 SYNCFREE
 SYNCRECEIVE
 SYNCSEND

Table 13. GDS ISSUE SIGNAL return codes

RETCODE (hexadecimal)	Description
03 00	CONVID is for a session that is not APPC.
03 00	CONVID is for a session that is in use by CPI Communications.
03 04	CONVID is for a conversation that is not basic.
03 08	A state check has occurred.
04	CONVID is for a session that is not allocated to task, or that is a relay link.

GDS RECEIVE

Receive data on an APPC basic conversation (assembler-language and C programs only).



Description

GDS RECEIVE receives data and indicators from a partner transaction.

The return code is given in RETCODE (see Table 14 on page 118). For a list of return code values, see the *CICS Distributed Transaction Programming Guide*. EXEC CICS conditions are never raised on GDS commands.

Options

In general, the arguments can be replaced by absolute or relocatable assembler-language expressions.

BUFFER

specifies that the length of the data passed to the application program in response to the RECEIVE command is to be restricted only by the length specified in the MAXFLENGTH option, and is not to be affected by GDS structured field boundaries. Control is returned to the application program when this length has been received, or when a synchronization request, change-direction, or end-bracket is received.

CONVDATA(data-area)

specifies the 24-byte application data area into which conversation-related information is to be moved. A description of the format of the data area is given in the discussion of CONVDATA fields in the *CICS Distributed Transaction Programming Guide*.

CONVID(name)

identifies the conversation to which the command relates. The 4-character name identifies either the token returned by a previously executed GDS ALLOCATE command, or the token representing the principal session (returned by a previously executed GDS ASSIGN command).

FLENGTH(data-area)

specifies a fullword binary data area that is set to the length of the data made available to the application program.

INTO(data-area)

specifies the application target data area into which data is to be received from the application program connected to the other end of the current conversation. The length of this area must not be less than the value specified in the MAXFLENGTH option.

LLID

specifies that the delimiter to be used by CICS to terminate the passing of data to the application program is the end of a GDS structured field, if this occurs before the MAXFLENGTH limit is reached.

MAXFLENGTH(data-value)

specifies, as a fullword binary value, either the length of the target data area specified in the INTO option, or the maximum length of data to be addressed by the pointer reference specified in the SET option. The length must not exceed 32 767 bytes. CICS does not receive more data than the MAXFLENGTH value allows.

RETCODE(data-area)

specifies the 6-byte application data area into which return code information is to be moved.

SET(pointer-ref)

specifies the pointer reference to be set to the address of data received from the application program connected to the other end of the current conversation. The pointer reference, unless changed by other commands or statements, is valid until the next RECEIVE (GDS or APPC) command, or the end of the task.

If DATALOCATION(ANY) is associated with the application program, the address of the data can be above or below the 16MB line.

If DATALOCATION(BELOW) is associated with the application program, and the data resides above the 16MB line, the data is copied below the 16MB line, and the address of this copy is returned.

If TASKDATAKEY(USER) is specified for the running task, and storage protection is active, the data returned is in a user-key. If TASKDATAKEY(CICS) is specified and storage protection is active, the data returned is in a CICS-key.

GDS RECEIVE

STATE(*cvda*)

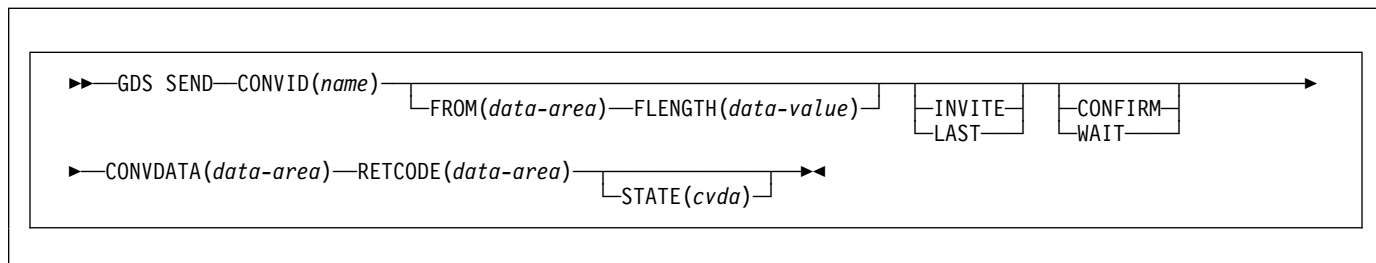
gets the state of the current conversation. The *cvda* values returned by CICS are:

ALLOCATED
CONFFREE
CONFRECEIVE
CONFSEND
FREE
PENDFREE
PENDRECEIVE
RECEIVE
ROLLBACK
SEND
SYNCFREE
SYNCRECEIVE
SYNCSEND

RETCODE (hexadecimal)	Description
03 00	CONVID is for a session that is not APPC.
03 00	CONVID is for a session that is in use by CPI Communications.
03 04	CONVID is for a conversation that is not basic.
03 08	A state check occurred.
04	CONVID is for a session that is not allocated to the task, or that is a relay link.
05 00 00 00 7F FF	MAXFLENGTH is outside the range 0 through 32 767.

GDS SEND

Send data on an APPC basic conversation (assembler-language and C programs only).



Description

GDS SEND sends data.

The return code is given in RETCODE (see Table 15 on page 120). For a list of return code values, see the *CICS Distributed Transaction Programming Guide*. EXEC CICS conditions are never raised on GDS commands.

Options

In general, the arguments can be replaced by absolute or relocatable assembler-language expressions.

CONFIRM

allows an application working at synchronization level 1 or 2 to synchronize its processing with that of a process in a remote system. The actions taken to synchronize processing are defined by the application programs involved. The CONFIRM option causes RQD2 to be added to the data already sent, and forces a WAIT. On receipt of the indicator, the remote process takes the agreed actions and then sends a response. When the WAIT completes, CDBERR is set to X'00' if the appropriate response has been received.

CONVDATA(data-area)

specifies the 24-byte application data area into which conversation-related information is to be moved. A description of the format of the data area is given in the discussion of CONVDATA fields in the *CICS Distributed Transaction Programming Guide*.

CONVID(name)

identifies the conversation to which the command relates. The 4-character name identifies either the token returned by a previously executed GDS ALLOCATE command, or the token representing the principal session (returned by a previously executed GDS ASSIGN command).

FLENGTH(data-value)

specifies the length (as a fullword binary value in the range 1–32 767) of the data specified in the FROM option.

FROM(data-area)

specifies the data that is to be sent.

INVITE

allows an application program to add a change-direction indicator to data already sent to a process in a connected APPC system. Control data is not transmitted by CICS until the subsequent execution of a WAIT or a SYNCPOINT command, unless CONFIRM or WAIT is also coded on the GDS SEND INVITE command.

LAST

allows an application program to add CEB to data already sent to a process in a connected APPC system. CEB is not transmitted by CICS until the subsequent execution of a WAIT or a SYNCPOINT command, unless CONFIRM or WAIT is also coded on the GDS SEND LAST command. Note that if one of these commands fails because of a conversation-related error, the conversation remains in bracket. In such a case, the application program should execute a GDS RECEIVE command. However, GDS SEND LAST WAIT (with no data) always causes the conversation to be deallocated.

RETCODE(data-area)

specifies the 6-byte application data area into which return code information is to be moved.

STATE(cvda)

gets the state of the current conversation. The cvda values returned by CICS are:

```

    ALLOCATED
    CONFFREE
    CONFRECEIVE
    CONFSEND
    FREE
    PENDFREE
    PENDRECEIVE
  
```

GDS SEND

RECEIVE
ROLLBACK
SEND
SYNCFREE
SYNCRECEIVE
SYNCSEND

WAIT

ensures that all data and indicators so far sent on a conversation are erased from the partner transaction.

If the WAIT option is not used, data from successive SEND commands is accumulated by CICS, together with any indicators, in an internal buffer. If the buffer becomes full, most of the accumulated data is transmitted to the remote system, but the accumulated indicators are not. Transmission of the accumulated data plus the indicators is forced by the WAIT or CONFIRM options of the GDS SEND command, or by a GDS WAIT command.

RETCODE (hexadecimal)	Description
03 00	CONVID is for a session that is not APPC.
03 00	CONVID is for a session that is in use by CPI Communications.
03 04	CONVID is for a conversation that is not basic.
03 08	A state check has occurred.
03 14	The CONFIRM option has been used on a sync level 0 conversation.
03 10	LL error (incorrect or incomplete).
04	CONVID is for a session that is not allocated to the task, or that is a relay link.
05 00 00 00 7F FF	The FLENGTH value is outside the range 0 through 32 767.

GDS WAIT

Ensure accumulated data transmitted on an APPC conversation (assembler-language and C programs only).

```

▶▶ GDS WAIT—CONVID(name)—CONVDATA(data-area)—RETCODE(data-area)—STATE(cvda)—▶▶

```

Description

GDS WAIT ensures that the accumulated data has been sent.

The return code is given in RETCODE (see Table 16). For a list of return code values, see the *CICS Distributed Transaction Programming Guide*. EXEC CICS conditions are never raised on GDS commands.

Options

In general, the arguments can be replaced by absolute or relocatable assembler-language expressions.

CONVDATA(*data-area*)

specifies the 24-byte application data area into which conversation-related information is to be moved. A description of the format of the data area is given in the discussion of CONVDATA fields in the *CICS Distributed Transaction Programming Guide*.

CONVID(*name*)

identifies the conversation to which the command relates. The 4-character name identifies either the token returned by a previously executed GDS ALLOCATE command, or the token representing the principal session (returned by a previously executed GDS ASSIGN command).

RETCODE(*data-area*)

specifies the 6-byte application data area into which return code information is to be moved.

STATE(*cvda*)

gets the state of the current conversation. The *cvda* values returned by CICS are:

```

ALLOCATED
CONFFREE
CONFRECEIVE
CONFSEND
FREE
PENDFREE
PENDRECEIVE
RECEIVE
ROLLBACK

```

```

SEND
SYNCFREE
SYNCRECEIVE
SYNCSSEND

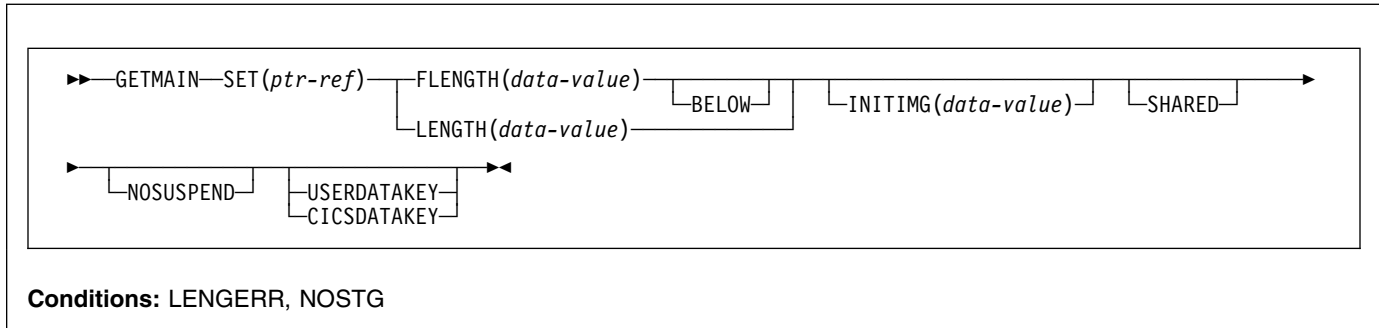
```

Table 16. GDS WAIT return codes

RETCODE (hexadecimal)	Description
03 00	CONVID is for a session that is not APPC.
03 00	CONVID is for a session that is in use by CPI Communications.
03 04	CONVID is for a conversation that is not basic.
03 08	A state check has occurred.
04	CONVID is for a session that is not allocated to task, or that is a relay link.

GETMAIN

Get main storage.



Dynamic transaction routing

Using GETMAIN with SHARED could create inter-transaction affinities that adversely affect the use of dynamic transaction routing. See the *CICS Application Programming Guide* for more information about transaction affinities.

Description

GETMAIN gets a main storage area of the size indicated by the FLENGTH option. (You can also use the LENGTH option, but this is supported for compatibility purposes and you are strongly recommended to use FLENGTH.) The address of the area is returned in the pointer reference supplied in the SET option.

CICS always allocates on double-word boundaries and rounds the requested length up to the nearest double-word multiple. Because there is no default initialization, you must use the INITIMG option if you require the storage to be initialized to a specific bit configuration.

CICS allocates storage from one of six different dynamic storage areas (DSAs):

- The CICS dynamic storage area (CDSA), below the 16MB line
- The user dynamic storage area (UDSA), below the 16MB line
- The shared dynamic storage area (SDSA), below the 16MB line
- The extended CICS dynamic storage area (ECDSA), above the 16MB line
- The extended user dynamic storage area (EUDSA), above the 16MB line
- The extended shared dynamic storage area (ESDSA), above the 16MB line

Note: There are two other dynamic storage areas—the read-only DSA (RDSA) and the extended read-only DSA (ERDSA)—but you cannot GETMAIN storage from these DSAs.

CICS determines whether to obtain the requested storage above or below the 16MB line, from one of the CICS- or user-key DSAs, or from one of the shared DSAs, according to the following options:

- The FLENGTH option with BELOW also specified
- The FLENGTH option alone, in conjunction with the addressing mode of the requesting program
- The LENGTH option
- The SHARED option

The above- or below-the-line allocation is summarized in the following table:

Options	Address mode	
	24-bit	31-bit
FLENGTH BELOW	A DSA	A DSA
FLENGTH alone	A DSA	An EDSA
LENGTH	A DSA	A DSA

CICS decides whether to allocate storage from a CICS-key or user-key DSA, or from a shared DSA, according to the following options:

- The USERDATAKEY option on the GETMAIN command
- The CICSDATAKEY option on the GETMAIN command
- The TASKDATAKEY option on the RDO TRANSACTION resource definition for the task under which the requesting program is running, if the USERDATAKEY and CICSDATAKEY options are both omitted
- The SHARED option on the GETMAIN command

The data-key option on the GETMAIN command overrides the TASKDATAKEY option on the RDO TRANSACTION resource definition. The effect of the data-key options,

without the SHARED option, is summarized in the following table:

Without SHARED option		
No data-key option	USERDATAKEY specified	CICSDATAKEY specified
Determined by TASKDATAKEY on TRANSACTION definition	User-key storage (from UDSA or EUDSA)	CICS-key storage (from CDSA or ECDSA)

The effect of the SHARED option is summarized in the following table:

With SHARED option		
No data-key option	USERDATAKEY specified	CICSDATAKEY specified
Determined by TASKDATAKEY on TRANSACTION definition	User-key storage (from SDSA or ESDSA)	CICS-key storage (from CDSA or ECDSA)

The storage that a task gets is available until it is released with a FREEMAIN command. For an area obtained without the SHARED option, only the task that acquired the storage may release it, and at task end CICS automatically releases such storage not already released. Note that any storage acquired with the SHARED option is accessible by all tasks.

A SHARED area, on the other hand, is not released at task end and remains until explicitly freed; any task may issue the FREEMAIN. This means that you can use SHARED storage in task-to-task communication.

You cannot, however, use the storage obtained as a TIOA for subsequent terminal operations, because this could cause storage violations.

Specifying CICSDATAKEY ensures that the requesting program obtains CICS-key storage from a CICS DSA, even if TASKDATAKEY(USER) is specified on the RDO TRANSACTION resource definition.

Options

BELOW

specifies that storage is to be obtained below the 16MB line, that is, from the CICS DSA.

CICSDATAKEY

specifies that CICS is to allocate storage from one of the CICS-key DSAs (the CDSA or ECDSA), overriding the TASKDATAKEY option specified on the RDO TRANSACTION resource definition. If you do not specify the data key, CICS determines the type of storage (CICS-key or user-key) according to the TASKDATAKEY option on the RDO TRANSACTION resource definition.

LENGTH(*data-value*)

specifies the number of bytes of storage required, in fullword binary format.

The maximum length that you can specify is the value of the corresponding DSA limit parameter, either DSALIMIT or EDSALIMIT. These are the system initialization parameters that define the overall storage limit within which CICS can allocate and manage the individual DSAs.

If the length requested is bigger than the DSALIMIT or EDSALIMIT value, the LENGERR condition occurs. If it is not bigger than these limits, but is more than is available, NOSTG occurs.

INITIMG(*data-value*)

specifies an optional 1-byte initialization value. If you specify INITIMG, CICS sets every byte of the acquired storage to the bit string you provide. Otherwise, CICS does not initialize the storage. In COBOL programs only, you must use a data area rather than a data value to define the initialization bit string.

LENGTH(*data-value*)

specifies the number of bytes (unsigned halfword binary value) of storage required. LENGTH implies storage from below the 16MB line and has an upper limit of 65 520 bytes. If you want storage above the 16MB line or a larger area, use FLENGTH.

If LENGTH is equal to zero, LENGERR occurs. If it is greater than the amount of storage available, NOSTG occurs.

Note: FLENGTH, with or without BELOW, is the recommended option: the LENGTH option is supported for compatibility purposes for those programs written to run under earlier releases of CICS.

NOSUSPEND

prevents CICS from suspending the task if no storage is available, and causes it to issue the NOSTG condition instead.

Note, however, that if a HANDLE CONDITION for NOSTG is active when the command is executed, control is passed to the user label supplied in the HANDLE CONDITION. This takes precedence over the NOSUSPEND option but is, of course, negated by either NOHANDLE or RESP.

SET(*ptr-ref*)

sets the pointer reference to the address of the acquired main storage. The pointer is set to the first byte of the storage area.

SHARED

prevents the automatic release of storage obtained by a GETMAIN command at the end of the task that requested it. This enables task-to-task communication. An area obtained with SHARED is not released until a corresponding FREEMAIN is issued, whether by the requesting task or some other task.

GETMAIN

Be aware that if a task abends, any shared storage acquired is not automatically released.

USERDATAKEY

specifies that CICS is to allocate storage from one of the user-key DSAs (the UDSA, SDSA, EUDSA, or ESDSA), overriding the TASKDATAKEY option specified on the RDO TRANSACTION resource definition. If you do not specify the data key, CICS determines the type of storage (CICS-key or user-key) according to the TASKDATAKEY option on the RDO TRANSACTION resource definition.

```
EXEC CICS GETMAIN SET(PTR)
          FLENGTH(2048)
          INITIMG(BLANK)
          CICSUSERKEY
```

Conditions

LENGERR

RESP2 values:

- 1 The FLENGTH value is less than 1 or greater than the length of the target storage area from which the request is to be satisfied. See the discussion about DSAs on page 122.

also occurs if the LENGTH value is zero.

Default action: terminate the task abnormally.

NOSTG

RESP2 values:

- 2 The storage requested is more than is currently available in the target DSA. See the discussion about DSAs on page 122.

Default action: ignore the condition. An active HANDLE CONDITION NOSTG also raises this condition.

Examples

The following example shows how to get a 1024-byte area from user-key storage below the 16MB line (assuming that TASKDATAKEY(USER) is specified on the RDO TRANSACTION resource definition), and initialize it to spaces:

```
EXEC CICS GETMAIN SET(PTR)
          FLENGTH(1024)
          BELOW
          INITIMG(BLANK)
```

You must define BLANK in your program as the character representing a space.

The following example shows how to get a 2048-byte area from CICS-key storage above the 16MB line (regardless of the TASKDATAKEY option specified on the RDO TRANSACTION resource definition), and initialize it to spaces:

GN (DL/I)

Get next segment.

```

>> GN [ USING PCB(integer-expr) ] [ KEYFEEDBACK(data-area) ] [ FEEDBACKLEN(integer-expr) ] <<

```

Condition:

For each parent segment:

```

>> [ VARIABLE ] [ FIRST ] [ SEGMENT(name) ] [ SEGLENGTH(integer-expr) ] >
    [ LAST ]
> [ OFFSET(integer-expr) ] <<

```

Condition:

For the object segment:

```

>> [ LOCKED ] [ INTO(data-area) ] [ WHERE(when-clause) ] [ FIELDLENGTH(integer-expr) ] <<

```

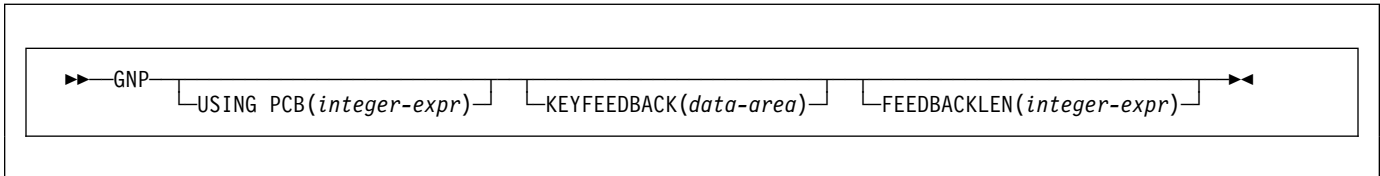
Description

The full form of the command is GET NEXT. See *DL/I DOS/VS Application Programming: High Level Programming Interfaces* for details.

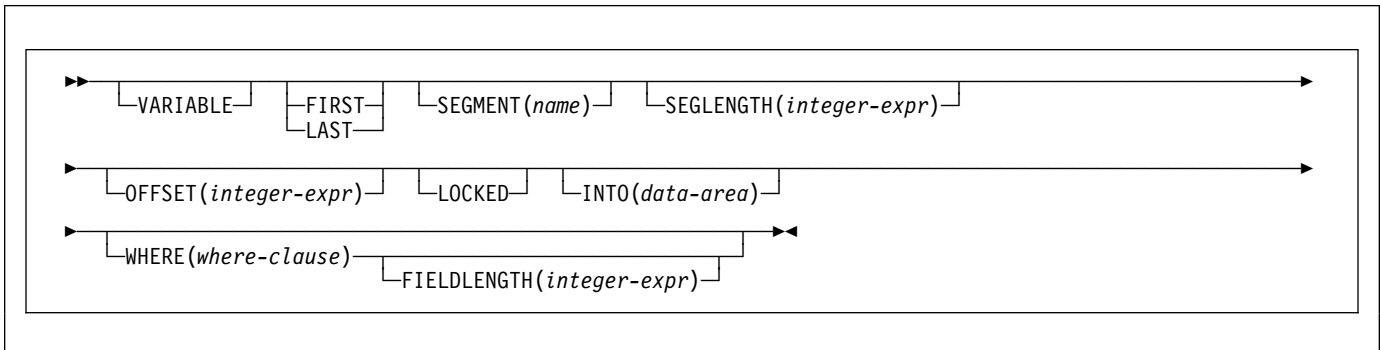
GNP (DL/I)

GNP (DL/I)

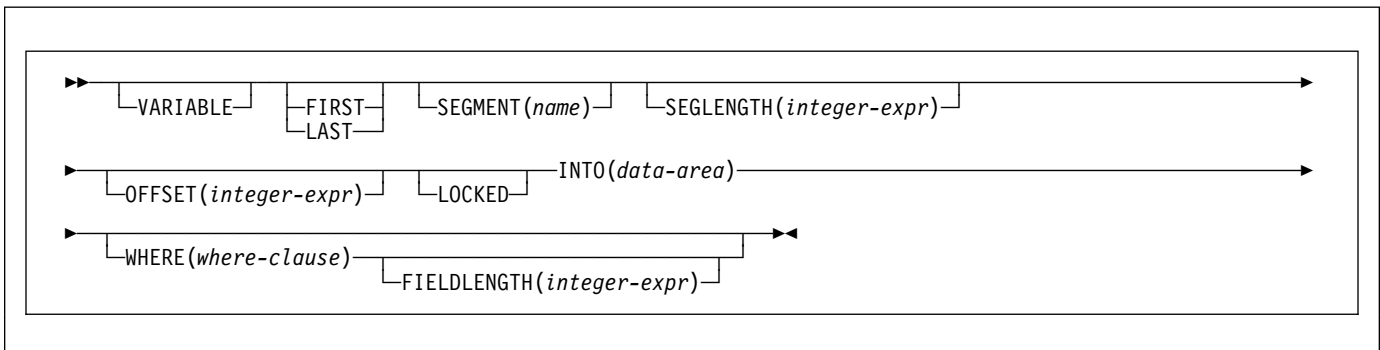
Get one or more segments.



For each parent segment:



For the object segment:

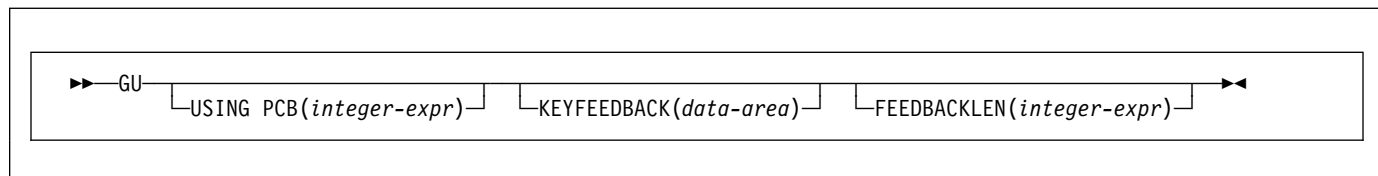


Description

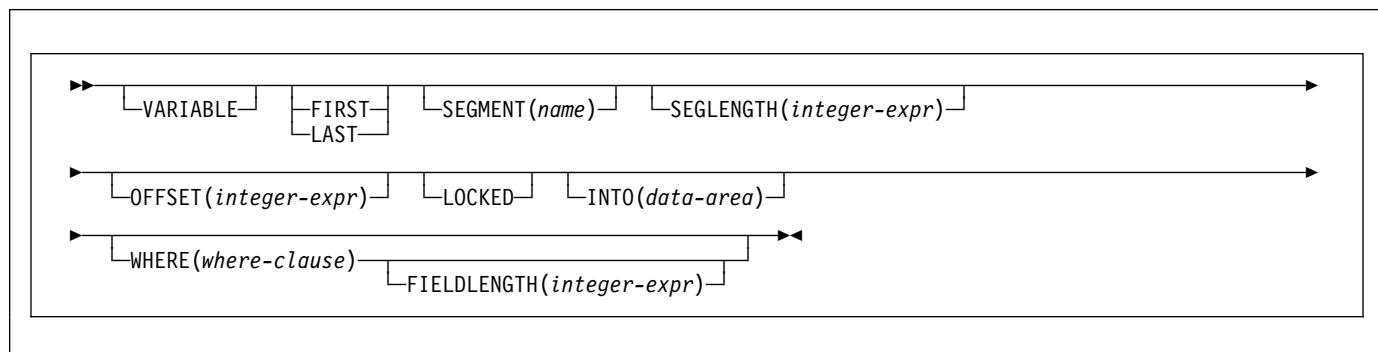
The full form of the command is GET IN PARENT. See *DL/I DOS/VS Application Programming: High Level Programming Interfaces* for details.

GU (DL/I)

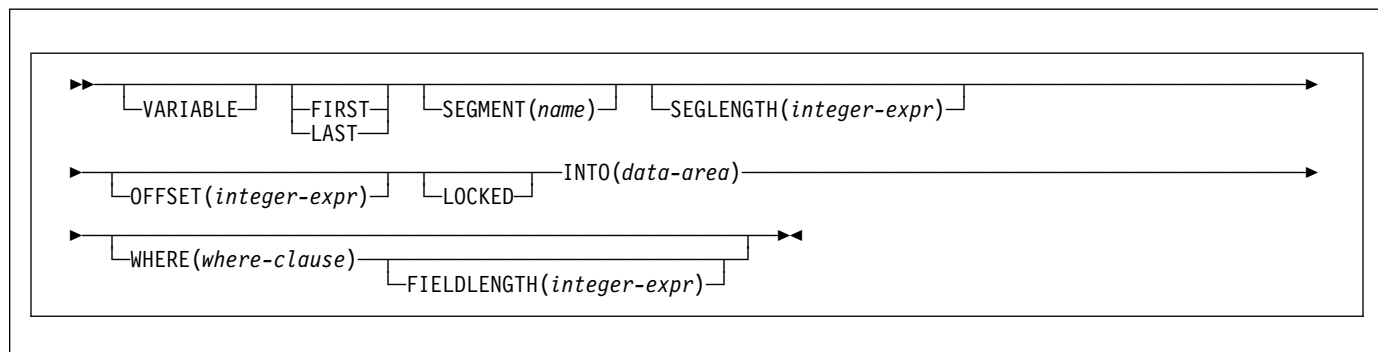
Get a segment.



For each parent segment:



For the object segment:

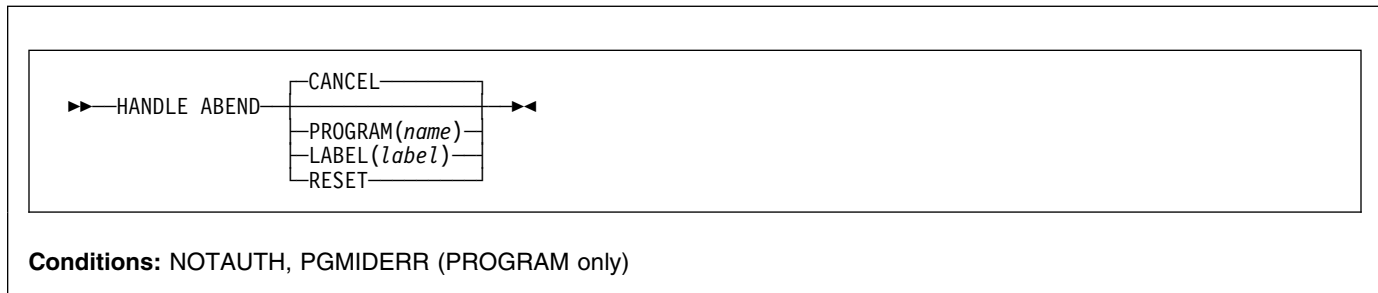


Description

The full form of the command is GET UNIQUE. See *DL/I DOS/VS Application Programming: High Level Programming Interfaces* for details.

HANDLE ABEND

Handle an abnormal termination exit.



Description

HANDLE ABEND is used to activate, cancel, or reactivate an exit for abnormal termination processing. You can suspend the command by means of the PUSH HANDLE and POP HANDLE commands as described in the *CICS Application Programming Guide*.

When a task terminates abnormally, CICS searches for an active abend exit, starting at the logical level of the application program in which the abend occurred, and proceeding to successively higher levels. The first active abend exit found, if any, is given control.

The HANDLE ABEND command cannot intercept abends that are issued with the CANCEL option. Some internal abends generated by CICS are issued with the CANCEL option, for example the ASPx or APSJ abend codes.

When the label specified in a HANDLE ABEND LABEL command receives control, the registers are set as follows:

COBOL: Control returns to the HANDLE ABEND command with the registers restored; COBOL GO TO statement is then executed.

ASM: R15 - Abend label.
R0-14 - Contents at the time of last CICS service request.

If LABEL is specified, the addressing mode is set to the addressing mode in which the HANDLE ABEND command has been issued; if PROGRAM is specified, the addressing mode is that defined by the way the program is link-edited.

If LABEL is specified, the execution key that the label receives control in, is the execution key that the program was running in when the HANDLE ABEND command was issued.

If PROGRAM is specified, the execution key is as specified by the EXECKEY option on the program's resource definition.

Options

CANCEL

specifies that a previously established exit at the logical level of the application program in control is to be canceled.

LABEL(*label*)

specifies the program label to which control branches if abnormal termination occurs.

This option cannot be used for C or PL/I application programs.

PROGRAM(*name*)

specifies the name of the program to which control is to be passed if the task is terminated abnormally. If this program has not already been defined, the program will be autoinstalled in the event of the abend condition being raised.

The program named in this option should always terminate with an abend, except when handling abends generated as a result of application program logic.

RESET

specifies that an exit canceled by a HANDLE ABEND CANCEL command, or by CICS, is to be reactivated.

This option is usually issued by an abnormal termination exit routine.

Conditions

NOTAUTH

occurs when a resource security check has failed on PROGRAM(*name*).

Default action: terminate the task abnormally.

PGMIDERR

RESP2 values:

- 1 The program has no entry in the PPT and autoinstall for programs is not active.
- 2 The program is disabled.
- 9 The installed program definition is for a remote program.

Default action: terminate the task abnormally.

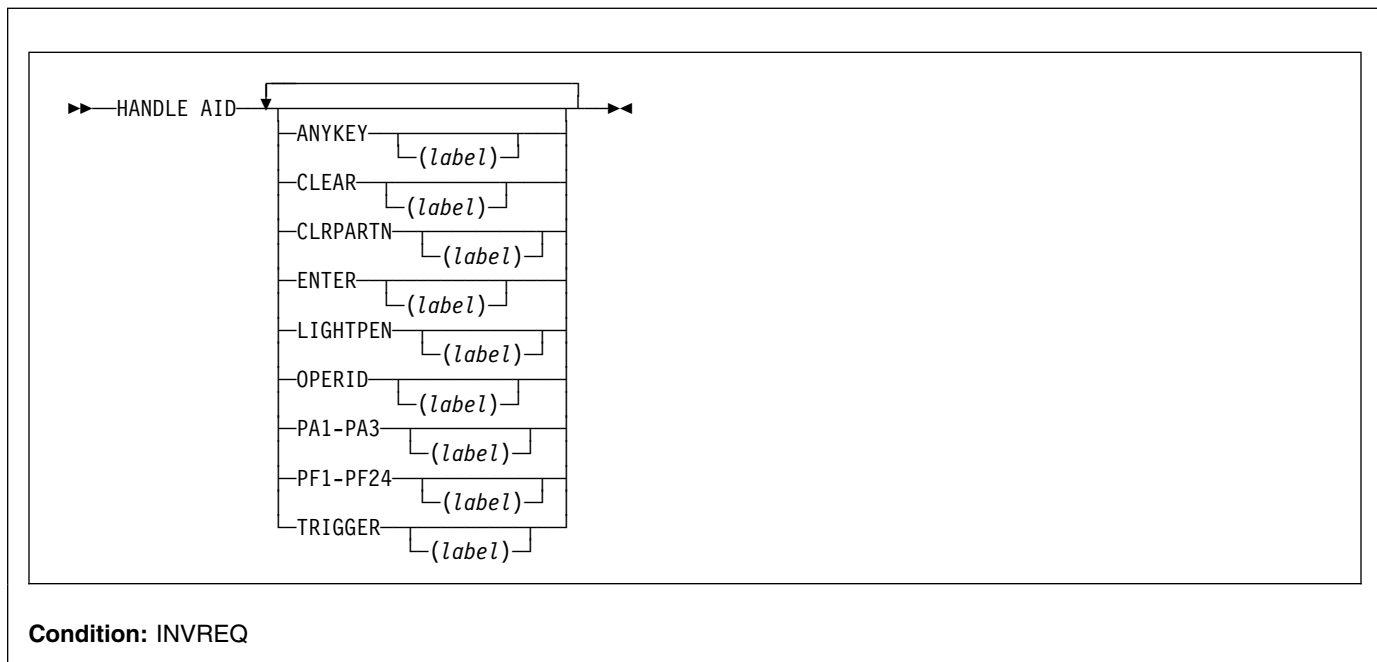
Examples

The following example shows how to establish a program as an exit:

```
EXEC CICS HANDLE ABEND  
PROGRAM('EXITPGM')
```

HANDLE AID

Handle attention identifiers (AIDs).



Description

HANDLE AID is used to specify the label to which control is to be passed when an AID is received from a display device. Control is passed after the input command is completed; that is, after any data received in addition to the AID has been passed to the application program.

To cause an AID to be ignored, issue a HANDLE AID command that specifies the associated option *without* a label. This deactivates the effect of that option in any previously-issued HANDLE AID command.

If no HANDLE AIDs are in effect, that is none have been issued or all have been canceled, control returns to the application program at the instruction immediately following the input command. Look in EIBAID to determine which key was pressed.

No more than 16 options are allowed in the same command.

The C language does not support HANDLE AID.

The options that can be specified are:

- ANYKEY (any PA key, any PF key, or the CLEAR key, but not ENTER)
- CLEAR (for the key of that name)
- CLRPARTN (for the key of that name)
- ENTER (for the key of that name)

- LIGHTPEN (for a light-pen attention)
- OPERID (for the operator identification card reader, the magnetic slot reader (MSR), or the extended MSR (MSRE))
- PA1, PA2, or PA3 (any of the program access keys)
- PF1 through PF24 (any of the program function keys)
- TRIGGER (for a trigger field attention)

If a task is initiated from a terminal by means of an AID, the first RECEIVE command in the task does not read from the terminal but copies only the input buffer (even if the length of the data is zero) so that control may be passed by means of a HANDLE AID command for that AID.

For the standard attention identifier list (DFHAID), and the standard attribute and printer control character list (DFHBMSCA), see Appendix J, "BMS-related constants" on page 405.

The execution key that the label receives control in, is the execution key that the program was running in when the HANDLE AID command was issued.

A print key specified by the system PRINT initialization parameter takes precedence over a HANDLE AID command.

Conditions

INVREQ

RESP2 values:

200 The command was issued by a distributed program link server application.

Default action: terminate the task abnormally.

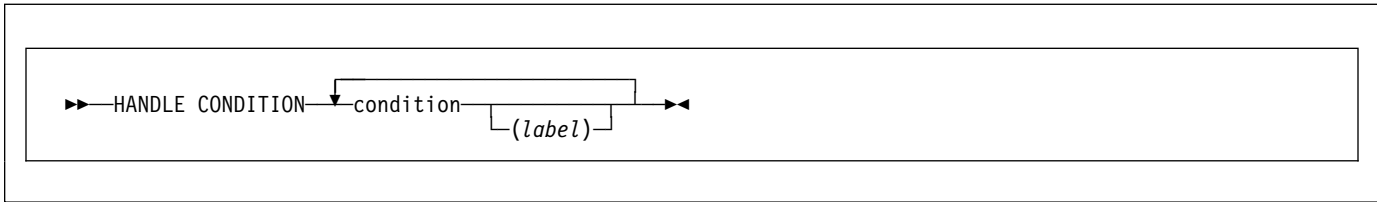
Examples

The following example shows a HANDLE AID command that specifies one label for the PA1 key; and a second label for CLEAR, PA2, PA3, and all the PF keys except PF10. If a PF10 AID is received or ENTER is pressed, control returns to the application program at the instruction immediately following the input command.

```
EXEC CICS HANDLE AID PA1(LAB1)
      ANYKEY(LAB2) PF10
```

HANDLE CONDITION

Handle conditions.



Description

You use HANDLE CONDITION to specify the label to which control is to be passed if a condition occurs. You must include the name of the condition and, optionally, a label to which control is to be passed if the condition occurs.

If you omit “label”, any HANDLE CONDITION command for the condition is deactivated, and the default action is taken if the condition occurs. This is independent of the setting of the generalized ERROR condition.

You must ensure that the HANDLE CONDITION command is executed before the command that may give rise to the associated condition.

You cannot include more than sixteen conditions in the same command; the conditions should be separated by at least one space. You must specify additional conditions in further HANDLE CONDITION commands.

If a condition occurs that is not specified in a HANDLE CONDITION or IGNORE CONDITION command, the default action is taken. If, however, the default action for a condition not specified in a HANDLE CONDITION or IGNORE CONDITION command terminates the task abnormally, and the condition ERROR has been specified, the action for ERROR is taken.

The execution key that the label receives control in, is the execution key that the program was running in when the HANDLE CONDITION command was issued.

Scope

The HANDLE CONDITION command for a given condition applies only to the program in which it is specified. The HANDLE CONDITION command:

- Remains active while the program is being executed, or until:
 - An IGNORE CONDITION command for the same condition is encountered, in which case the HANDLE CONDITION command is overridden

- Another HANDLE CONDITION command for the same condition is encountered, in which case the new command overrides the previous one.
- A LINK command is executed to call another CICS program. The HANDLE CONDITION options are not inherited by the linked-to program.
- Is temporarily deactivated by the NOHANDLE or RESP option on a command.

Language considerations

In an assembler-language application program, if a HANDLE CONDITION and the command that causes the condition are at the same logical level, the registers are restored to their values in the application program at the point where the command that caused the condition was issued. If, however, the command that causes a condition occurs at a lower logical level, the registers are restored to their current values saved in DFHEISTG at the HANDLE CONDITION level.

In a PL/I application program, a branch to a label in an inactive procedure or in an inactive begin block, caused by a condition, produces unpredictable results.

The C language does not support HANDLE CONDITION.

Options

condition(label)

specifies the name of the condition; “label” specifies the location within the program to be branched to if the condition occurs.

For more information about the conditions, see Appendix A, “EXEC interface block” on page 373.

Examples

The following example shows you how to handle conditions, such as DUPREC, LENGERR, and so on, that can occur when you use a WRITE command to add a record to a data set.

Suppose that you want DUPREC to be handled as a special

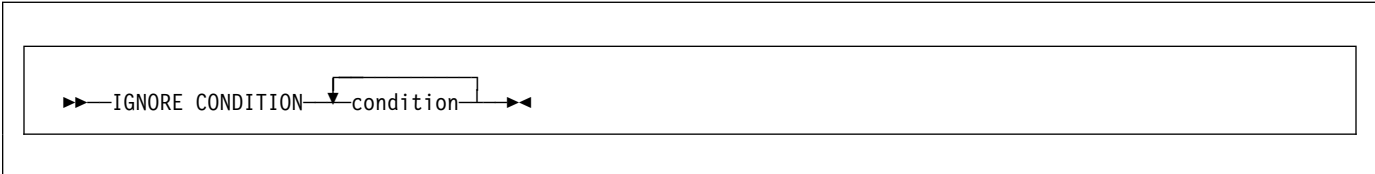
case; that you want default action (that is, to terminate the task abnormally) to be taken for LENGERR; and that you want all other conditions to be handled by the error routine ERRHANDL. You would code:

```
EXEC CICS HANDLE CONDITION  
        ERROR(ERRHANDL)  
        DUPREC(DUPRTN) LENGERR
```

IGNORE CONDITION

IGNORE CONDITION

Ignore conditions.



Description

IGNORE CONDITION is not supported for C programs.

For information about the conditions, see Appendix A, "EXEC interface block" on page 373.

You use IGNORE CONDITION to specify that no action is to be taken if a condition occurs (that is, control returns to the instruction following the command that has failed to execute, and the EIB is set). Execution of a command could result in several conditions being raised. CICS checks these in a predetermined order and only the first one that is not ignored (by your IGNORE CONDITION command) is passed to your application program.

The IGNORE CONDITION command for a given condition applies only to the program in which it is specified, and it remains active while the program is being executed, or until a HANDLE CONDITION command for the same condition is encountered, in which case the IGNORE CONDITION command is overridden.

You cannot include more than sixteen conditions in the same command; the conditions must be separated by at least one space. You may specify additional conditions in further IGNORE CONDITION commands.

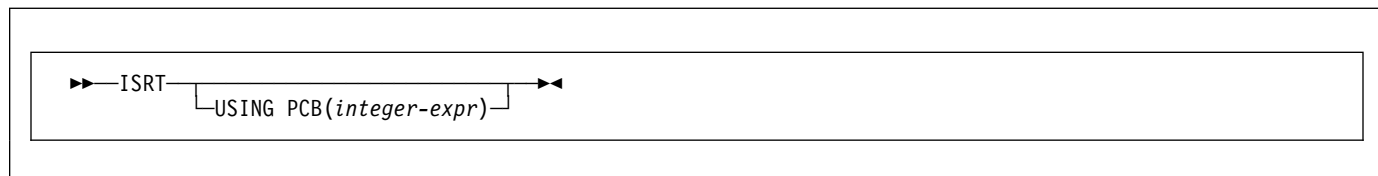
Options

condition

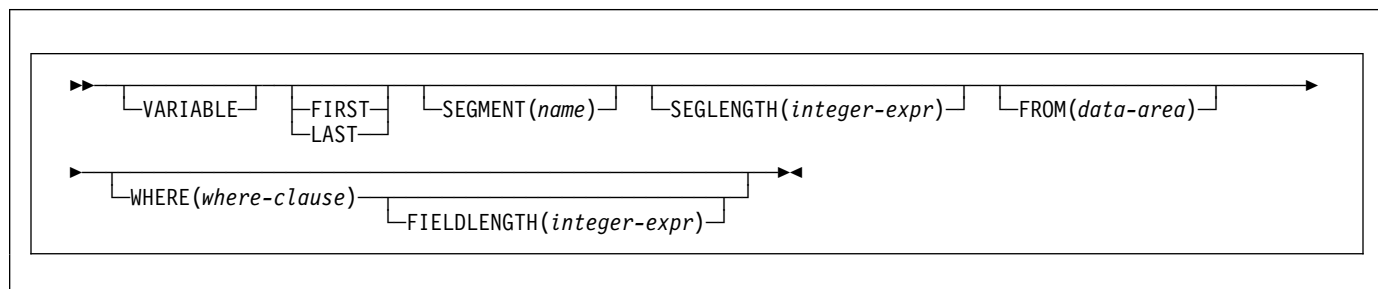
specifies the name of the condition to be ignored.

ISRT (DL/I)

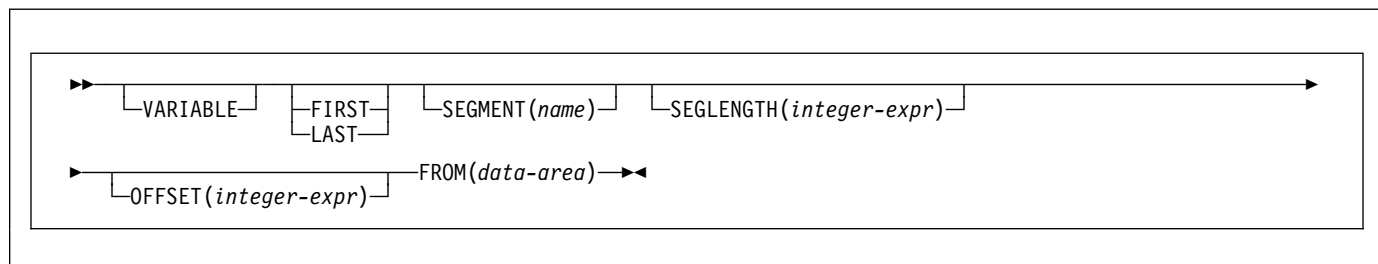
Insert one or more segments.



For each parent segment:



For the object segment:

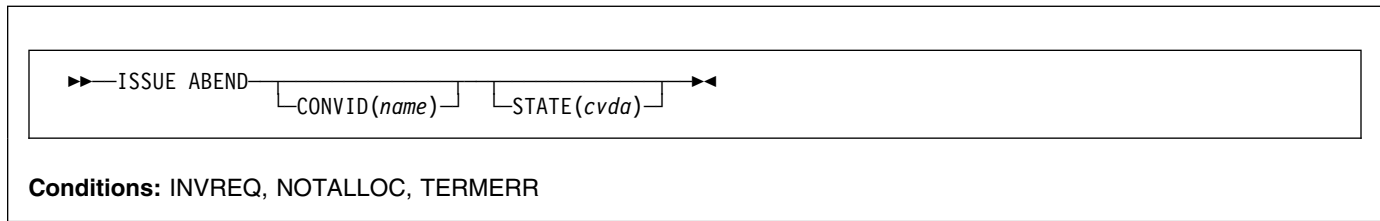


Description

The full form of the command is INSERT. See *DL/I DOS/VS Application Programming: High Level Programming Interfaces* for details.

ISSUE ABEND

Abend the mapped conversation with an APPC partner.



Description

ISSUE ABEND abnormally ends the conversation. The partner transaction sees the TERMERR condition.

Options

CONVID(*name*)

identifies the conversation to be abended. The 4-character name identifies either the symbolic identifier returned by a previously executed ALLOCATE command in EIBRSRCE in the EIB, or the symbolic identifier representing the principal facility (returned by a previously executed ASSIGN command).

For compatibility with earlier releases, SESSION is accepted as a synonym for CONVID. New programs should use CONVID.

If both CONVID and SESSION are omitted, the principal facility is assumed.

STATE(*cvda*)

gets the state of the current conversation. The *cvda* values returned by CICS are:

ALLOCATED
 CONFREE
 CONFRECEIVE
 CONFSEND
 FREE
 PENDFREE
 PENDRECEIVE
 RECEIVE
 ROLLBACK
 SEND
 SYNCFREE
 SYNCRECEIVE
 SYNCSEND

Conditions

INVREQ

RESP2 values:

- 200** A distributed program link server application specified the function-shipping session (its principal facility) on the CONVID option.

also occurs (RESP2 not set) in any of the following situations:

- ISSUE ABEND is used on any conversation other than an EXEC CICS APPC mapped conversation.

Default action: terminate the task abnormally.

NOTALLOC

occurs if the specified CONVID value relates to a conversation that is not owned by the application.

Default action: terminate the task abnormally.

TERMERR

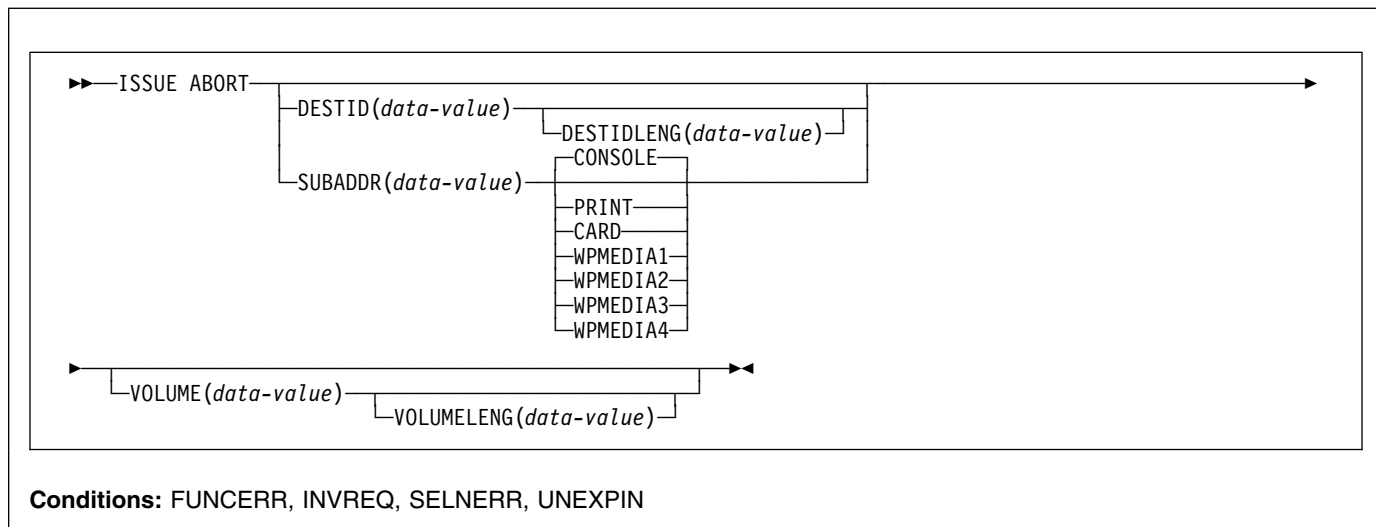
occurs for a session-related error. Any action on that conversation other than a FREE command causes an ATCV abend.

A CANCEL TASK request by a user node error program (NEP) may cause this condition if the task has an outstanding terminal control request active when the node abnormal condition program handles the session error.

Default action: terminate the task abnormally with abend code ATNI.

ISSUE ABORT

End processing of a data set abnormally.



Description

ISSUE ABORT ends communication with a data set in an outboard controller, or the selected medium, abnormally. The data set specified in the **DESTID** option is deselected abnormally. The options **CONSOLE**, **PRINT**, **CARD**, and **WPMEDIA1–4** are alternatives to **DESTID** and **DESTIDLENG**.

Options

CARD

specifies that the output medium is a card reader or card punch device. This option is not valid with **DESTID** and **DESTIDLENG**.

CONSOLE

specifies that the output medium is that provided for messages to the operator. This option is not valid with **DESTID** and **DESTIDLENG**. This refers to a programmable subsystem such as the IBM 3790 data communication system. It does not refer to a CICS or system console.

DESTID(data-value)

specifies the name (1–8 characters) of the data set in the outboard destination.

DESTIDLENG(data-value)

specifies the length (halfword binary value) of the name specified in the **DESTID** option.

PRINT

specifies that the output medium is a printer.

SUBADDR(data-value)

specifies the medium subaddress as a halfword binary value (in the range 0 through 15) which allows media of the same type, for example, “printer 1” or “printer 2”, to be defined. Value 15 means a medium of any type. The default is zero.

VOLUME(data-value)

specifies the name (1–6 characters) of a diskette in an outboard destination that contains the data set specified in the **DESTID** option.

VOLUMELENG(data-value)

specifies the length (halfword binary value) of the name specified in the **VOLUME** option.

WPMEDIA1 through WPMEDIA4

specifies that, for each specific LUTYPE4 device, a word-processing medium is defined to relate to a specific input/output device.

Conditions

FUNCERR

occurs if there is an error during execution of the command. Destination selection is unaffected and other commands for the same destination may be successful.

Default action: terminate the task abnormally.

INVREQ

RESP2 values:

200 A distributed program link server application specified the function-shipping session (its principal facility) on the **CONVID** option.

ISSUE ABORT

Default action: terminate the task abnormally.

SELNERR

occurs if there is an error during destination selection. The destination is not selected and other commands for the same destination are unlikely to be successful.

Default action: terminate the task abnormally.

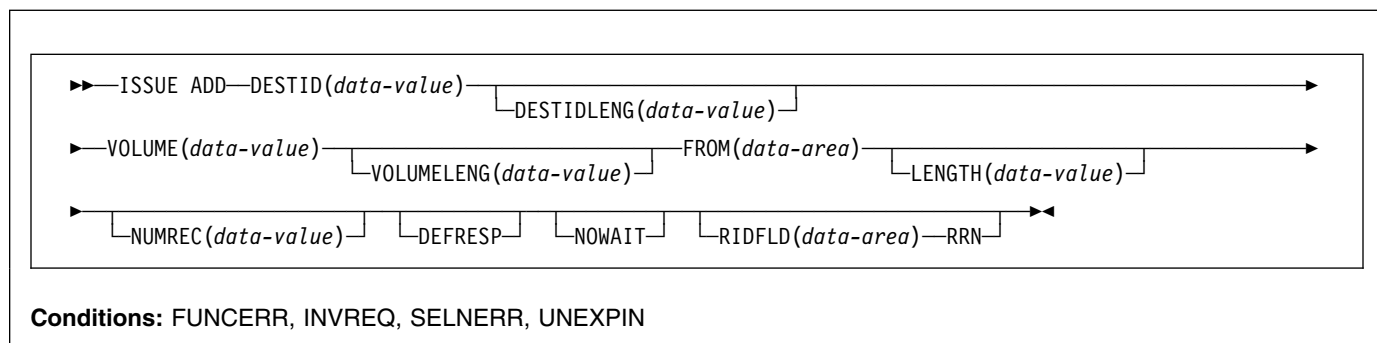
UNEXPIN

occurs when some unexpected or unrecognized information is received from the outboard controller.

Default action: terminate the task abnormally.

ISSUE ADD

Add a record to a data set.



Description

ISSUE ADD adds records to a sequential or keyed direct data set in an outboard controller. The FROM option is used to specify the data to be written, and the LENGTH option specifies its length.

The RIDFLD option is only needed with this command when it applies to a DPCX/DXAM data set. In this case, it specifies the relative record number of the record to be added. When RIDFLD is used, NUMREC must be 1 (the default).

Options

DEFRESP

specifies that all terminal control commands issued as a result of the ISSUE ADD command are to request a definite response from the outboard batch program, irrespective of the specification of message integrity for the CICS task (by the system programmer).

DESTID(*data-value*)

specifies the name (1–8 characters) of the data set in the outboard destination.

DESTIDLENG(*data-value*)

specifies the length (halfword binary value) of the name specified in the DESTID option.

FROM(*data-area*)

specifies the data to be written to the data set.

LENGTH(*data-value*)

specifies the length (halfword binary value) of the data to be written. For a description of a safe upper limit, see “LENGTH options” on page 5.

NOWAIT

specifies that the CICS task continues processing without waiting for the ISSUE ADD command to

complete. If this option is not specified, the task activity is suspended until the command is completed.

NUMREC(*data-value*)

for a relative record data set, specifies as a halfword binary value the number of logical records to be added. Records are replaced sequentially starting with the one identified by the RIDFLD option.

For an indexed data set, NUMREC cannot be specified because only one record can be added.

RIDFLD(*data-area*)

specifies, for a relative record data set, a 4-character field as the relative record number (starting from zero) of the record. The RRN option is also required.

For a keyed direct data set, RIDFLD should specify a key.

RRN

specifies that the record identification field specified in the RIDFLD option contains a relative record number. This option is required for a relative record data set.

VOLUME(*data-value*)

specifies the name (1–6 characters) of a diskette in an outboard destination that contains the data set specified in the DESTID option.

VOLUMELENG(*data-value*)

specifies the length (halfword binary value) of the name specified in the VOLUME option.

ISSUE ADD

Conditions

FUNCERR

occurs if there is an error during execution of the command. Destination selection is unaffected and other commands for the same destination may be successful.

Default action: terminate the task abnormally.

INVREQ

RESP2 values:

200 A distributed program link server application specified the function-shipping session (its principal facility) on the CONVID option.

SELNERR

occurs if there is an error during destination selection. The destination is not selected and other commands for the same destination are unlikely to be successful.

Default action: terminate the task abnormally.

UNEXPIN

occurs when some unexpected or unrecognized information is received from the outboard controller.

Default action: terminate the task abnormally.

ISSUE CONFIRMATION

Issue a positive response to a SEND CONFIRM on an APPC mapped conversation.

```

  >>—ISSUE CONFIRMATION—┌CONVID(name)┐┌STATE(cvda)┐—>>
  
```

Conditions: INVREQ, NOTALLOC, SIGNAL, TERMERR

Description

ISSUE CONFIRMATION allows an application program to respond positively when the CONFIRM option has been specified on a SEND command executed by a partner transaction.

Options

CONVID(*name*)

identifies the conversation in which to send the response. The 4-character name identifies either the token returned by a previously executed ALLOCATE command in EIBRSRCE in the EIB, or the token representing the principal facility (returned by a previously executed ASSIGN command).

For compatibility with earlier releases, SESSION is accepted as a synonym for CONVID. New programs should use CONVID.

If both CONVID and SESSION are omitted, the principal facility is assumed.

STATE(*cvda*)

gets the state of the current conversation. The *cvda* values returned by CICS are:

```

  ALLOCATED
  CONFFREE
  CONFRECEIVE
  CONFSEND
  FREE
  PENDFREE
  PENDRECEIVE
  RECEIVE
  ROLLBACK
  SEND
  SYNCFREE
  SYNCRECEIVE
  SYNCSEND
  
```

Conditions

INVREQ

RESP2 values:

- 200** A distributed program link server application specified the function-shipping session on the CONVID option.

also occurs (RESP2 not set) in any of the following situations:

- ISSUE CONFIRMATION is used on a conversation that is either of the following:
 - Sync level 0
 - Not APPC mapped

Default action: terminate the task abnormally.

NOTALLOC

occurs if the specified CONVID value relates to a conversation that is not owned by the application.

Default action: terminate the task abnormally.

SIGNAL

occurs when an inbound SIGNAL data-flow control command is received from a partner transaction. EIBSIG is always set when an inbound signal is received.

Default action: ignore the condition.

TERMERR

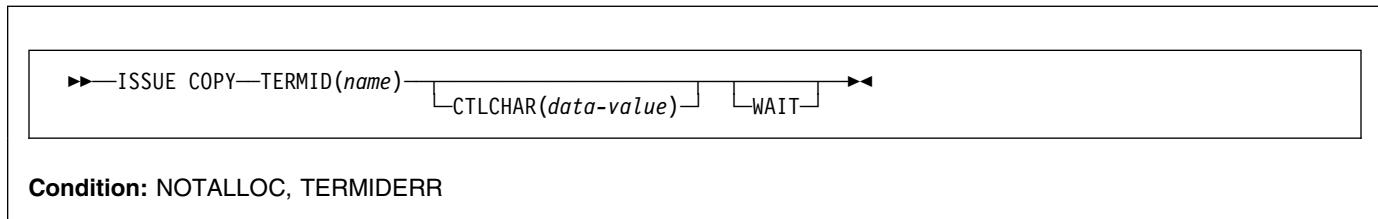
occurs for a session-related error. Any action on that conversation other than a FREE causes an ATCV abend.

A CANCEL TASK request by a user node error program (NEP) may cause this condition if the task has an outstanding terminal control request active when the node abnormal condition program handles the session error.

Default action: terminate the task abnormally with abend code ATNI.

ISSUE COPY (3270 display)

Copy data from 3270 information display system (BTAM).



Description

Default action: terminate the task abnormally.

ISSUE COPY copies the format and data contained in the buffer of a specified terminal into the buffer of the terminal that started the transaction. Both terminals must be attached to the same remote control unit.

Options

CTLCHAR(*data-value*)

specifies a 1-byte copy control character (CCC) that defines the copy function. A COBOL user must specify a data area containing this character. If the option is omitted, the contents of the entire buffer (including nulls) are copied.

TERMINID(*name*)

specifies the name (1–4 characters) of the terminal whose buffer is to be copied. The terminal must have been defined in the TCT.

WAIT

specifies that processing of the command must be completed before any subsequent processing is attempted.

If the WAIT option is not specified, control is returned to the application program when processing of the command has started. A subsequent input or output request (terminal control, BMS, or batch data interchange) to the terminal associated with the task causes the application program to wait until the previous request has been completed.

Conditions

NOTALLOC

occurs if the facility specified in the command is not owned by the application.

Default action: terminate the task abnormally.

TERMIDERR

occurs if the specified terminal identifier cannot be found in the terminal control table (TCT).

ISSUE COPY (3270 logical)

Copy data from 3270 logical unit.

```

▶▶—ISSUE COPY—TERMINID(name)—┌CTLCHAR(data-value)┐┌WAIT┐▶▶

```

Conditions: LENGERR, NOTALLOC, TERMERR

Description

ISSUE COPY copies the format and data contained in the buffer of a specified terminal into the buffer of the terminal that started the transaction. Both terminals must be attached to the same remote control unit.

Options

CTLCHAR(*data-value*)

specifies a 1-byte copy control character (CCC) that defines the copy function. A COBOL user must specify a data area containing this character. If the option is omitted, the contents of the entire buffer (including nulls) are copied.

TERMINID(*name*)

specifies the name (1–4 characters) of the terminal whose buffer is to be copied. The terminal must have been defined in the TCT.

WAIT

specifies that processing of the command must be completed before any subsequent processing is attempted.

If the WAIT option is not specified, control is returned to the application program once processing of the command has started. A subsequent input or output request (terminal control, BMS, or batch data interchange) to the terminal associated with the task causes the application program to wait until the previous request has been completed.

Conditions

LENGERR

occurs if an out-of-range value is supplied.

Default action: terminate the task abnormally.

NOTALLOC

occurs if the facility specified in the command is not owned by the application.

Default action: terminate the task abnormally.

TERMERR

occurs for a terminal-related error.

A CANCEL TASK request by a user node error program (NEP) may cause this condition if the task has an outstanding terminal control request active when the node abnormal condition program handles the session error.

Default action: terminate the task abnormally with abend code ATNI.

ISSUE DISCONNECT (default)

ISSUE DISCONNECT (default)

Terminate a session between CICS and a logical unit or terminal.

▶▶—ISSUE DISCONNECT—◀◀

Conditions: SIGNAL, TERMERR

Description

ISSUE DISCONNECT terminates sessions between CICS and the following terminals or logical units:

- 3270-display logical unit (LUTYPE2)
- 3270-printer logical unit (LUTYPE3)
- LUTYPE4 logical unit
- 3270 SCS printer logical unit
- System/7
- 2260 or 2265 display station
- 3270 logical unit
- 3600 pipeline logical unit
- 3600(3601) logical unit
- 3600(3614) logical unit
- 3630 plant communication system
- 3650 interpreter logical unit
- 3650 host conversational (3270) logical unit
- 3650 host conversational (3653) logical unit
- 3650(3680) host command processor logical unit
- 3735 programmable buffered terminal
- 3740 data entry system
- 3767/3770 interactive logical unit
- 3770 batch logical unit
- 3790 logical units

It occurs for a terminal-related error, such as a session failure. This condition applies to VTAM-connected terminals only. Because of the asynchronous nature of this condition, the application program should check, using SEND CONFIRM or SYNCPOINT, to make sure any errors still outstanding have been resolved before it relinquishes control. If you wish to handle this condition, you must first issue a FREE command to free the session. If you do not do this, an INVREQ condition occurs, plus an ATCV abend if you do not handle this condition.

A CANCEL TASK request by a user node error program (NEP) may cause this condition if the task has an outstanding terminal control request active when the node abnormal condition program handles the session error.

Default action: terminate the task abnormally with abend code ATNI.

Conditions

For most terminal and logical unit types, ISSUE DISCONNECT raises no conditions. Exceptions are:

SIGNAL

occurs only for an ISSUE DISCONNECT for LUTYPE4, 3600(3601), 3767 interactive, 3770 batch, and 3790 full-function logical units.

It occurs when an inbound SIGNAL data-flow control command is received from a logical unit or session. EIBSIG is always set when an inbound signal is received.

Default action: ignore the condition.

TERMERR

occurs only for an ISSUE DISCONNECT for LUTYPE4 logical units.

ISSUE DISCONNECT (LUTYPE6.1)

Disconnect an LUTYPE6.1 logical unit.

```

▶▶—ISSUE DISCONNECT—┐
                       └─SESSION(name)─┘▶▶
  
```

Conditions: NOTALLOC, TERMERR

Description

ISSUE DISCONNECT disconnects the unit, if DISCREQ(YES) is specified in the RDO TYPETERM resource definition.

Options

SESSION(*name*)

specifies the symbolic identifier (1–4 characters) of a session TCTTE. This option specifies the alternate facility to be disconnected. If this option is omitted, the principal facility for the task is disconnected.

Conditions

NOTALLOC

occurs if the facility specified in the command is not owned by the application.

Default action: terminate the task abnormally.

TERMERR

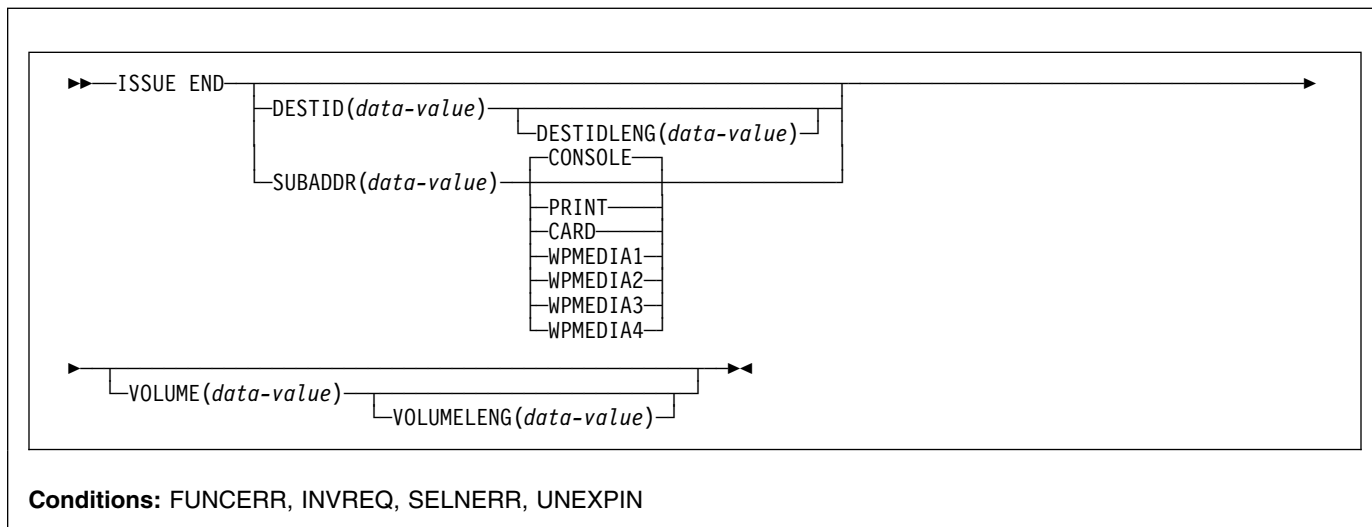
occurs for a terminal-related error, such as a session failure.

A CANCEL TASK request by a user node error program (NEP) may cause this condition if the task has an outstanding terminal control request active when the node abnormal condition program handles the session error.

Default action: terminate the task abnormally with abend code ATNI.

ISSUE END

End processing of a data set.



Description

ISSUE END ends communication with a data set in an outboard controller or with the selected medium. The data set specified in the DESTID option, or the selected medium, is deselected normally. The options CONSOLE, PRINT, CARD, and WPMEDIA1–4 are alternatives to DESTID and DESTIDLENG.

Options

CARD

specifies that the output medium is a card reader or card punch device. This option is not valid with DESTID and DESTIDLENG.

CONSOLE

specifies that the output medium is that provided for messages to the operator. This option is not valid with DESTID and DESTIDLENG. This refers to a programmable subsystem such as the IBM 3790 data communication system. It does not refer to a CICS or system console.

DESTID(*data-value*)

specifies the name (1–8 characters) of the data set in the outboard destination.

DESTIDLENG(*data-value*)

specifies the length (halfword binary value) of the name specified in the DESTID option.

PRINT

specifies that the output medium is a printer.

SUBADDR(*data-value*)

specifies the medium subaddress as a halfword binary value (in the range 0–15) that allows media of the same type, for example, “printer 1” or “printer 2”, to be defined. Value 15 means a medium of any type. The default is zero.

VOLUME(*data-value*)

specifies the name (1–6 characters) of a diskette in an outboard destination that contains the data set specified in the DESTID option.

VOLUMELENG(*data-value*)

specifies the length (halfword binary value) of the name specified in the VOLUME option.

WPMEDIA1 through WPMEDIA4

specifies that, for each specific LUTYPE4 device, a word-processing medium is defined to relate to a specific input/output device.

Conditions

FUNCERR

occurs if there is an error during the execution of the command. Destination selection is unaffected and other commands for the same destination may be successful.

Default action: terminate the task abnormally.

INVREQ

RESP2 values:

200 A distributed program link server application specified the function-shipping session (its principal facility) on the CONVID option.

Default action: terminate the task abnormally.

SELNERR

occurs if there is an error during destination selection. The destination is not selected and other commands for the same destination are unlikely to be successful.

Default action: terminate the task abnormally.

UNEXPIN

occurs when some unexpected or unrecognized information is received from the outboard controller.

Default action: terminate the task abnormally.

ISSUE ENDFILE

Indicate the end-of-file condition to the 3740 data entry system.

The diagram shows a command structure within a rectangular box. On the left, there are two right-pointing arrows followed by the text "ISSUE ENDFILE". A horizontal line extends from the end of "ISSUE ENDFILE" to the left side of a bracketed structure. This structure consists of a vertical line going down, a horizontal line going right, and another vertical line going down. Below this horizontal line is the text "ENDOUTPUT". A horizontal line extends from the right side of the bracketed structure to the right side of the box, ending with two left-pointing arrows.

Condition: INVREQ, NOTALLOC

Description

ISSUE ENDFILE indicates the end-of-file condition to the 3740.

Options

ENDOUTPUT

indicates the end-of-output condition as well as end of file.

Conditions

INVREQ

RESP2 values:

200 A distributed program link server application attempted to send on its function shipping session, its principal facility.

Default action: terminate the task abnormally.

NOTALLOC

occurs if the facility specified in the command is not owned by the application.

Default action: terminate the task abnormally.

ISSUE ENDOUTPUT

Indicate the end-of-output condition to the 3740 data entry system.



Description

ISSUE ENDOUTPUT indicates the end-of-output condition to the 3740.

Options

ENDFILE

indicates the end-of-file condition as well as end of output.

Conditions

INVREQ

RESP2 values:

200 A distributed program link server application attempted to send on its function shipping session, its principal facility.

Default action: terminate the task abnormally.

NOTALLOC

occurs if the facility specified in the command is not owned by the application.

Default action: terminate the task abnormally.

ISSUE EODS

Send end-of-data-set function management header to the 3650 interpreter logical unit.



▶—ISSUE EODS—◀

Conditions: INVREQ, NOTALLOC, TERMERR

Description

ISSUE EODS issues the end-of-data-set management header.

Conditions

INVREQ

RESP2 values:

200 A distributed program link server application attempted to send on its function shipping session, its principal facility.

Default action: terminate the task abnormally.

NOTALLOC

occurs if the facility specified in the command is not owned by the application.

Default action: terminate the task abnormally.

TERMERR

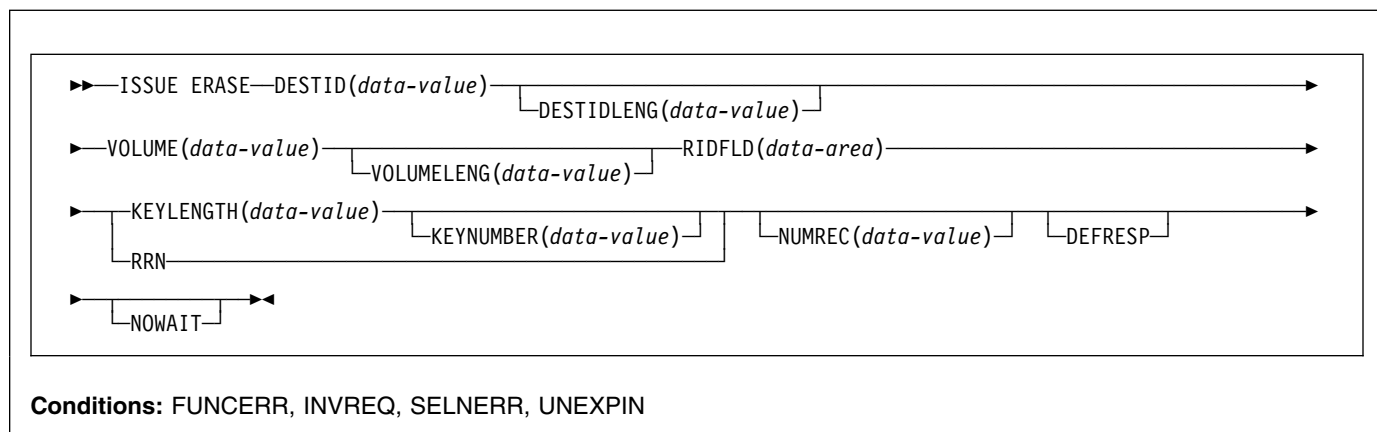
occurs for a terminal-related error, such as a session failure.

A CANCEL TASK request by a user node error program (NEP) may cause this condition if the task has an outstanding terminal control request active when the node abnormal condition program handles the session error.

Default action: terminate the task abnormally with abend code ATNI.

ISSUE ERASE

Delete a record from a data set.



Description

ISSUE ERASE deletes a record from a keyed direct data set in an outboard controller, or erases a record from a DPCX or DXAM relative record data set.

Options

DEFRESP

specifies that all terminal control commands issued as a result of the ISSUE ERASE command are to request a definite response from the outboard batch program, irrespective of the specification of message integrity for the CICS task (by the system programmer).

DESTID(*data-value*)

specifies the name (1–8 characters) of the data set in the outboard destination.

DESTIDLENG(*data-value*)

specifies the length (halfword binary value) of the name specified in the DESTID option.

KEYLENGTH(*data-value*)

specifies the length of the key specified in the RIDFLD option, as a halfword binary value.

KEYNUMBER(*data-value*)

specifies the number, as a halfword binary value, of the index to be used to locate the record. There can be eight indexes (1–8). The default is 1. This option applies only to DPCX or DXAM and is mutually exclusive with RRN.

NOWAIT

specifies that the CICS task continues processing without waiting for the ISSUE ERASE command to complete. If this option is not specified, the task activity is suspended until the command is completed.

NUMREC(*data-value*)

for a relative record data set, specifies as a halfword binary value the number of logical records to be deleted. Records are replaced sequentially starting with the one identified by the RIDFLD option.

For an indexed data set, NUMREC cannot be specified, because only one record is deleted.

RIDFLD(*data-area*)

specifies the record identification field.

For a relative record data set, the RIDFLD option specifies a fullword binary integer (the relative record number of the record starting from zero); and the RRN option is used.

For an indexed data set, the RIDFLD option specifies the key that is embedded in the data. The KEYLENGTH option is also required.

RRN

specifies that the record identification field specified in the RIDFLD option contains a relative record number. If the option is not specified, RIDFLD is assumed to specify a key.

VOLUME(*data-value*)

specifies the name (1–6 characters) of a diskette in an outboard destination that contains the data set specified in the DESTID option.

VOLUMELENG(*data-value*)

specifies the length (halfword binary value) of the name specified in the VOLUME option.

ISSUE ERASE

Conditions

FUNCERR

occurs if there is an error during execution of the command. Destination selection is unaffected and other commands for the same destination may be successful.

Default action: terminate the task abnormally.

INVREQ

RESP2 values:

200 A distributed program link server application specified the function-shipping session (its principal facility) on the CONVID option.

Default action: terminate the task abnormally.

SELNERR

occurs if there is an error during destination selection. The destination is not selected and other commands for the same destination are unlikely to be successful.

Default action: terminate the task abnormally.

UNEXPIN

occurs when some unexpected or unrecognized information is received from the outboard controller.

Default action: terminate the task abnormally.

ISSUE ERASEAUP

Erase all unprotected fields of a 3270 buffer.



Conditions: INVREQ, NOTALLOC, TERMERR

Description

ISSUE ERASEAUP erases unprotected fields by:

1. Clearing all unprotected fields to nulls (X'00')
2. Resetting modified data tags in each unprotected field to zero
3. Positioning the cursor to the first unprotected field
4. Restoring the keyboard

You can use the ISSUE ERASEAUP command for the following types of 3270 logical units:

- 3270-display logical unit (LUTYPE2)
- 3270-printer logical unit (LUTYPE3)
- 3270 logical unit
- 3650 host conversational (3270) logical unit
- 3790 (3270-display) logical unit
- 3790 (3270-printer) logical unit

Options

WAIT

ensures that the erase is completed before control returns to the application program. If you omit WAIT, control returns to the application program as soon as ISSUE ERASEAUP starts processing.

Conditions

INVREQ

RESP2 values:

- 200** A distributed program link server application specified the function-shipping session (its principal facility) on the CONVID option.

Default action: terminate the task abnormally.

NOTALLOC

occurs if the facility specified in the command is not owned by the application.

Default action: terminate the task abnormally.

TERMERR

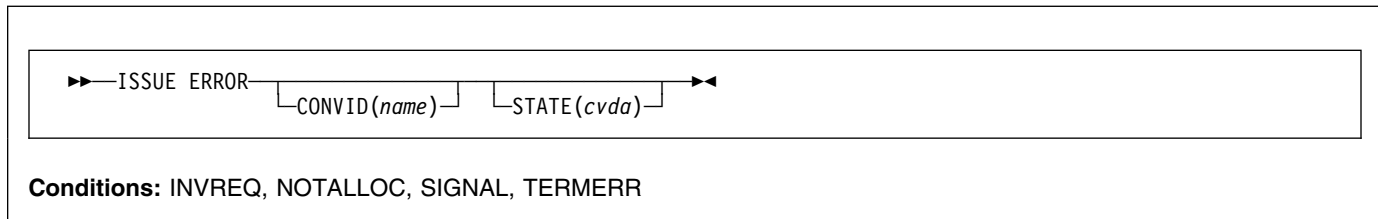
occurs if there is a terminal-related error.

A CANCEL TASK request by a user node error program (NEP) may cause this condition if the task has an outstanding terminal control request active when the node abnormal condition program handles the session error.

Default action: terminate the task abnormally with abend code ATNI.

ISSUE ERROR

Inform APPC mapped conversation partner of error.



Description

ISSUE ERROR allows an application program to inform a process in a connected APPC system that some program-detected error has occurred. For example, a remote CICS application is notified by having EIBERR set, with EIBERRCD=X'0889'. The actions required to recover from the error are the responsibility of logic contained in both application programs. The application program can use this command to respond negatively when the CONFIRM option has been specified on a SEND command executed by a process in a connected APPC system.

Options

CONVID(*name*)

identifies the conversation to which the command relates. The 4-character name identifies either the token returned by a previously executed ALLOCATE command in EIBRSRCE in the EIB, or the token representing the principal facility (returned by a previously executed ASSIGN command).

For compatibility with earlier releases, SESSION is accepted as a synonym for CONVID. New programs should use CONVID.

If both CONVID and SESSION are omitted, the principal facility is assumed.

STATE(*cvda*)

gets the state of the current conversation. The *cvda* values returned by CICS are:

ALLOCATED
 CONFFREE
 CONFRECEIVE
 CONFSEND
 FREE
 PENDFREE
 PENDRECEIVE
 RECEIVE
 ROLLBACK
 SEND
 SYNCFREE
 SYNCRECEIVE

SYNCSEND

Conditions

INVREQ

RESP2 values:

- 200** A distributed program link server application specified the function-shipping session on the CONVID option.

also occurs (RESP2 not set) in any of the following situations:

- The command is not valid for the APPC conversation type in use.
- The command is issued against a CPI-Communications conversation.

Default action: terminate the task abnormally.

NOTALLOC

occurs if the specified CONVID value does not relate to a conversation owned by the application.

Default action: terminate the task abnormally.

SIGNAL

occurs when an inbound SIGNAL data-flow control command is received from a partner transaction. EIBSIG is always set when an inbound signal is received.

Default action: ignore the condition.

TERMERR

occurs for a session-related error. Any action on that conversation other than a FREE command causes an ATCV abend.

A CANCEL TASK request by a user node error program (NEP) may cause this condition if the task has an outstanding terminal control request active when the node abnormal condition program handles the session error.

Default action: terminate the task abnormally with abend code ATNI.

ISSUE LOAD

Specify the name of a program on 3650 interpreter logical unit.

```
▶▶—ISSUE LOAD—PROGRAM(name)—┌CONVERSE┐▶▶
```

Conditions: NONVAL, NOTALLOC, NOSTART, TERMERR

Description

ISSUE LOAD specifies the name of the 3650 application program that is to be loaded.

Options

CONVERSE

specifies that the 3650 application program is able to communicate with the host processor. If this option is not specified, the 3650 application program cannot communicate with the host processor.

PROGRAM(*name*)

specifies the name (1–8 characters) of the 3650 application program that is to be loaded.

Conditions

NONVAL

occurs if the 3650 application program name is not valid.
Default action: terminate the task abnormally.

NOSTART

occurs if the 3651 is unable to initiate the requested 3650 application program.
Default action: terminate the task abnormally.

NOTALLOC

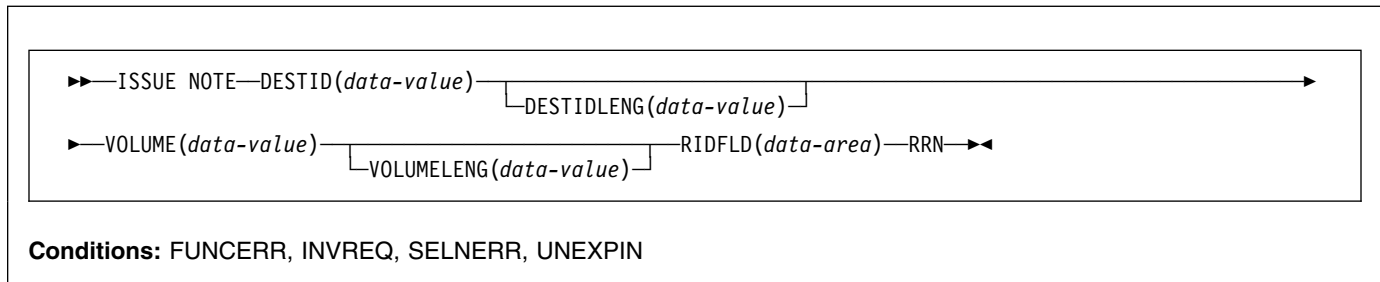
occurs if the facility specified in the command is not owned by the application.
Default action: terminate the task abnormally.

TERMERR

occurs for a terminal-related error.
A CANCEL TASK request by a user node error program (NEP) may cause this condition if the task has an outstanding terminal control request active when the node abnormal condition program handles the session error.
Default action: terminate the task abnormally with abend code ATNI.

ISSUE NOTE

Request next record number.



Description

ISSUE NOTE requests the number of the next record. It finds the relative record number of the next record in an addressed direct data set. The number is returned in the data area specified in the RIDFLD option. The RRN option must be specified, because a relative record number is involved.

Options

DESTID(*data-value*)

specifies the name (1–8 characters) of the data set in the outboard destination.

DESTIDLENG(*data-value*)

specifies the length (halfword binary value) of the name specified in the DESTID option.

RIDFLD(*data-area*)

specifies as a 4-character field a data area the relative record number of the next record is returned in.

RRN

specifies that the record identification field specified in the RIDFLD option contains a relative record number.

VOLUME(*data-value*)

specifies the name (1–6 characters) of a diskette in an outboard destination that contains the data set specified in the DESTID option.

VOLUMELENG(*data-value*)

specifies the length (halfword binary value) of the name specified in the VOLUME option.

Conditions

FUNCERR

occurs if there is an error during execution of the command. Destination selection is unaffected and other commands for the same destination may be successful.

Default action: terminate the task abnormally.

INVREQ

RESP2 values:

200 A distributed program link server application specified the function-shipping session (its principal facility) on the CONVID option.

Default action: terminate the task abnormally.

SELNERR

occurs if there is an error during destination selection. The destination is not selected and other commands for the same destination are unlikely to be successful.

Default action: terminate the task abnormally.

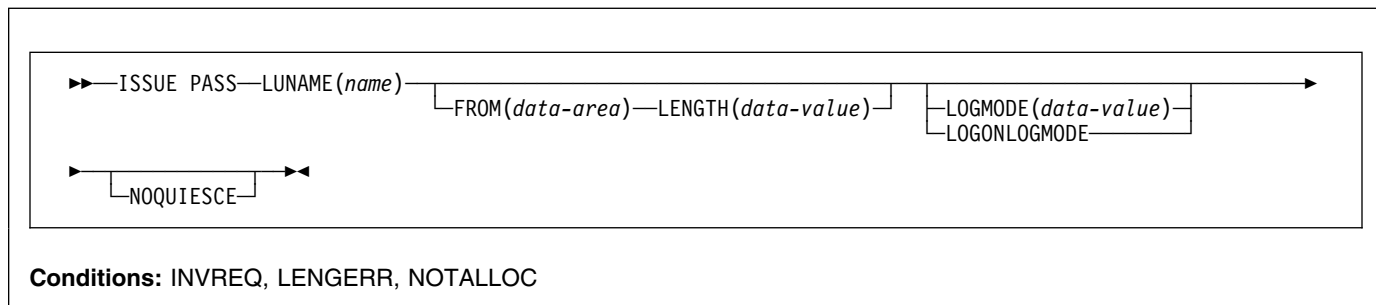
UNEXPIN

occurs when some unexpected or unrecognized information is received from the outboard controller.

Default action: terminate the task abnormally.

ISSUE PASS

VTAM application routing.



Description

ISSUE PASS disconnects the terminal from CICS after the task has terminated, and transfers it to the VTAM application defined in the LUNAME option.

This command requires that AUTH=PASS is coded on the VTAM APPL macro for the CICS terminal-owning system that issues it, with DISCREQ(YES) or RELREQ(YES) specified in the RDO TYPETERM resource definition for any terminal where this function might be used.

If the LUNAME specified is the name of another CICS system, you can use the EXTRACT LOGONMSG command to access the data referred to by this command.

Because of a VTAM limitation, the maximum length of the user data is restricted to 255 bytes.

Note: The system initialization parameter CLSDSTP=NOTIFYINONOTIFY allows you to have the node error program (NEP) and the console notified of whether the PASS was successful or not. The NEP can be coded to reestablish a session ended by an unsuccessful PASS. For programming information about how to do this, see the section about the NEP in the *CICS Customization Guide*.

Options

FROM(*data-area*)

specifies the data area containing the logon user data that is to be passed to the application named in the LUNAME option. This option may be omitted if ATTACHID is specified on an LUTYPE6.1 command.

LENGTH(*data-value*)

specifies the length, as a halfword binary value, of the data issued.

LOGMODE(*data-value*)

specifies the name (1–8 characters) of the VTAM logon mode table entry used by VTAM to establish the new session.

LOGONLOGMODE

specifies that the new session is to be established with the VTAM logon mode table entry in use when the session logged on.

Note: The logmode name saved is taken from the X'0D' control vector in the VTAM CINIT. This is the logmode name known in this system.

If persistent sessions is in use (The System Initialization Parameter PSDINT=nnn has been specified), then the RDO TYPETERM resource definition for any terminal to be ISSUE PASSEd should use RECOVOPTION(NONE), because the logon LOGMODE name is not recovered across a persistent sessions restart.

If neither LOGMODE nor LOGONLOGMODE is supplied, the new session will be established with the default LOGMODE.

LUNAME(*name*)

specifies the name (1–8 characters) of the VTAM application to which the terminal is to be passed.

NOQUIESCE

specifies that the user can choose to recover from certain pass failures.

Conditions

INVREQ

occurs if the command is not valid for the logical unit in use.

Default action: terminate the task abnormally.

LENGERR

occurs if an out-of-range value is supplied in the LENGTH option.

Default action: terminate the task abnormally.

ISSUE PASS

NOTALLOC

occurs if the facility specified in the command is not owned by the application.

Default action: terminate the task abnormally.

ISSUE PREPARE

Issue the first flow of a syncpoint request on an APPC mapped conversation.

```

  >>—ISSUE PREPARE—┐┐CONVID(name)┐┐STATE(cvda)┐┐
  
```

Conditions: INVREQ, NOTALLOC, TERMERR

Description

ISSUE PREPARE applies only to distributed transaction processing over APPC links. It enables a syncpoint initiator to prepare a syncpoint slave for syncpointing by sending only the first flow (prepare-to-commit) of the syncpoint exchange. Depending on the reply from the syncpoint slave, the initiator can proceed with the syncpoint by issuing a SYNCPOINT command, or initiate back-out by issuing a SYNCPOINT ROLLBACK command.

Options

CONVID(*name*)

identifies the conversation to which the command relates. The 4-character name identifies either the token returned by a previously executed ALLOCATE command in EIBRSRCE in the EIB, or the token representing the principal facility (returned by a previously executed ASSIGN command).

If this option is omitted, the principal facility is assumed.

STATE(*cvda*)

gets the state of the current conversation. The *cvda* values returned by CICS are:

```

  ALLOCATED
  CONFREE
  CONFRECEIVE
  CONFSEND
  FREE
  PENDFREE
  PENDRECEIVE
  RECEIVE
  ROLLBACK
  SEND
  SYNCFREE
  SYNCRECEIVE
  SYNCSEND
  
```

Conditions

INVREQ

RESP2 values:

- 200** A distributed program link server application specified the function-shipping session (its principal facility) on the CONVID option.

also occurs (RESP2 not set) in any of the following situations:

- The conversation is not an APPC mapped conversation.
- The conversation state is not valid for the request.
- The sync level of the conversation is not 2.

Default action: terminate the task abnormally.

NOTALLOC

occurs if the CONVID value in the command does not relate to a conversation that is owned by the application.

Default action: terminate the task abnormally.

TERMERR

occurs for a session-related error. Any action on that conversation other than a FREE causes an ATCV abend.

A CANCEL TASK request by a user node error program (NEP) may cause this condition if the task has an outstanding terminal control request active when the node abnormal condition program handles the session error.

Default action: terminate the task abnormally with abend code ATNI.

ISSUE PRINT

Print displayed data on first available printer.

▶▶—ISSUE PRINT—◀◀

Conditions: INVREQ, NOTALLOC, TERMERR

Description

ISSUE PRINT prints displayed data on the first available printer that can respond to a print request.

ISSUE PRINT can be used on a number of logical units, using the printers defined below:

- For a 3270 logical unit or a 3650 host conversational (3270) logical unit, the printer must be defined by the PRINTER or ALTPRINTER options of the RDO TERMINAL resource definition, or by a printer supplied by the autoinstall user program.
- For a 3270-display logical unit with the printer adapter feature, used with a 3274 or 3276, the printer is allocated by the printer authorization matrix. The printer adapter feature is enabled by specifying DEVICE(LUTYPE2) and PRINTADAPTER(YES) on the RDO TYPETERM resource definition.
- For a 3790 (3270-display) logical unit, the printer is allocated by the 3790.

The printer must be in service, not currently attached to a task, and owned by the same CICS system that owns the terminal running the transaction.

outstanding terminal control request active when the node abnormal condition program handles the session error.

Default action: terminate the task abnormally with abend code ATNI.

Conditions

INVREQ

RESP2 values:

- 200** A distributed program link server application specified the function-shipping session (its principal facility) on the CONVID option.

Default action: terminate the task abnormally.

NOTALLOC

occurs if the facility specified in the command is not owned by the application.

Default action: terminate the task abnormally.

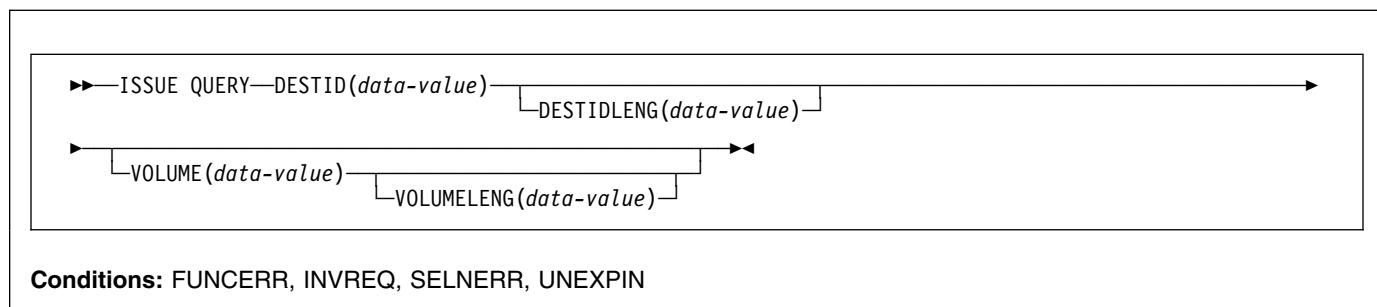
TERMERR

occurs for a terminal-related error.

A CANCEL TASK request by a user node error program (NEP) may cause this condition if the task has an

ISSUE QUERY

Interrogate a data set.



Description

ISSUE QUERY interrogates a data set. It is used to request that a sequential data set in an outboard controller be transmitted to the host system. The application program should either follow this command with ISSUE RECEIVE commands to get the resulting inbound data, or terminate the transaction to allow CICS to start a new transaction to process the data.

Options

DESTID(*data-value*)

specifies the name (1–8 characters) of the data set in the outboard destination.

DESTIDLENG(*data-value*)

specifies the length (halfword binary value) of the name specified in the DESTID option.

VOLUME(*data-value*)

specifies the name (1–6 characters) of a diskette in an outboard destination that contains the data set specified in the DESTID option.

VOLUMELENG(*data-value*)

specifies the length (halfword binary value) of the name specified in the VOLUME option.

Conditions

FUNCERR

occurs if there is an error during execution of the command. Destination selection is unaffected and other commands for the same destination may be successful.

Default action: terminate the task abnormally.

INVREQ

RESP2 values:

200 A distributed program link server application specified the function-shipping session (its principal facility) on the CONVID option.

Default action: terminate the task abnormally.

SELNERR

occurs if there is an error during destination selection. The destination is not selected and other commands for the same destination are unlikely to be successful.

Default action: terminate the task abnormally.

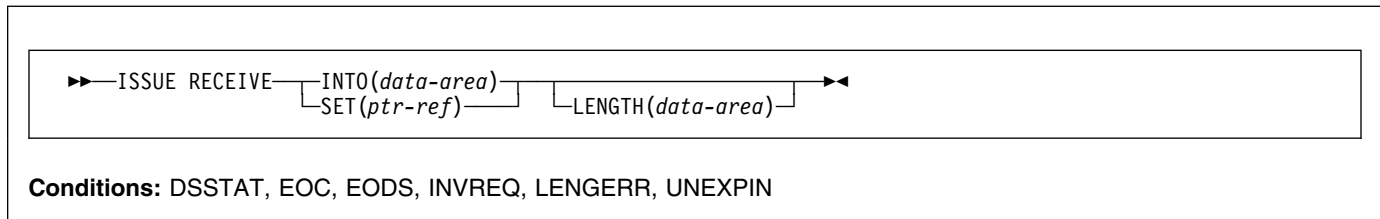
UNEXPIN

occurs when some unexpected or unrecognized information is received from the outboard controller.

Default action: terminate the task abnormally.

ISSUE RECEIVE

Read a record from a data set.



Description

ISSUE RECEIVE reads a sequential data set in an outboard controller.

The INTO option specifies the area into which the data is to be placed. The LENGTH option must specify a data area that contains the maximum length of record that the program accepts. If the record length exceeds the specified maximum length, the record is truncated and the LENGERR condition occurs. After the retrieval operation, the data area specified in the LENGTH option is set to the record length (before any truncation occurred).

Alternatively, a pointer reference can be specified in the SET option. CICS then acquires an area of sufficient size to hold the record, and sets the pointer reference to the address of that area. After the retrieval operation, the data area specified in the LENGTH option is set to the record length.

The outboard controller might not send the data from the data set specified in the ISSUE QUERY command. The ASSIGN command must be used to get the value of DESTID (which identifies the data set that has actually been transmitted) and the value of DESTIDLENG (which is the length of the identifier in DESTID).

Options

INTO(*data-area*)

specifies the receiving field for the data read from the data set.

If you specify the ISSUE RECEIVE command with the INTO option, the parameter must be a data area that specifies the maximum length of data that the program is prepared to handle. If the value specified is less than zero, zero is assumed. If the length of the data exceeds the value specified, the data is truncated to that value and the LENGERR condition occurs. On completion of the retrieval operation, the data area is set to the original length of the data.

LENGTH(*data-area*)

specifies the length (halfword binary value) of the data received.

If you have specified SET, you must also specify LENGTH.

SET(*ptr-ref*)

specifies the pointer reference that is to be set to the address location of the data read from the data set.

If you specify the SET option, the parameter must be a data area. On completion of the retrieval operation, the data area is set to the length of the data.

If DATALOCATION(ANY) is specified on the RDO PROGRAM resource definition for the application program, the address of the data can be above or below the 16MB line.

If DATALOCATION(BELOW) is specified, and the data resides above the 16MB line, the data is copied below the 16MB line, and the address of this copy is returned.

If TASKDATAKEY(USER) is specified on the RDO TRANSACTION resource definition for the running task, and storage protection is active, the data returned is in a user-key. If TASKDATAKEY(CICS) is specified and storage protection is active, the data returned is in a CICS-key.

If you have specified SET, you must also specify LENGTH.

Conditions

DSSTAT

occurs when the destination status changes in one of the following ways:

- The data stream is abended.
- The data stream is suspended.

Default action: terminate the task abnormally.

EOC

occurs if the request/response unit (RU) is received with the end-of-chain (EOC) indicator set. Field EIBEOC also contains this indicator.

Default action: ignore the condition.

EODS

occurs when the end of the data set is encountered.

Default action: terminate the task abnormally.

INVREQ

RESP2 values:

200 A distributed program link server application specified the function-shipping session (its principal facility) on the CONVID option.

Default action: terminate the task abnormally.

LENGERR

occurs if the length of the retrieved data is greater than the value specified by the LENGTH option.

Default action: terminate the task abnormally.

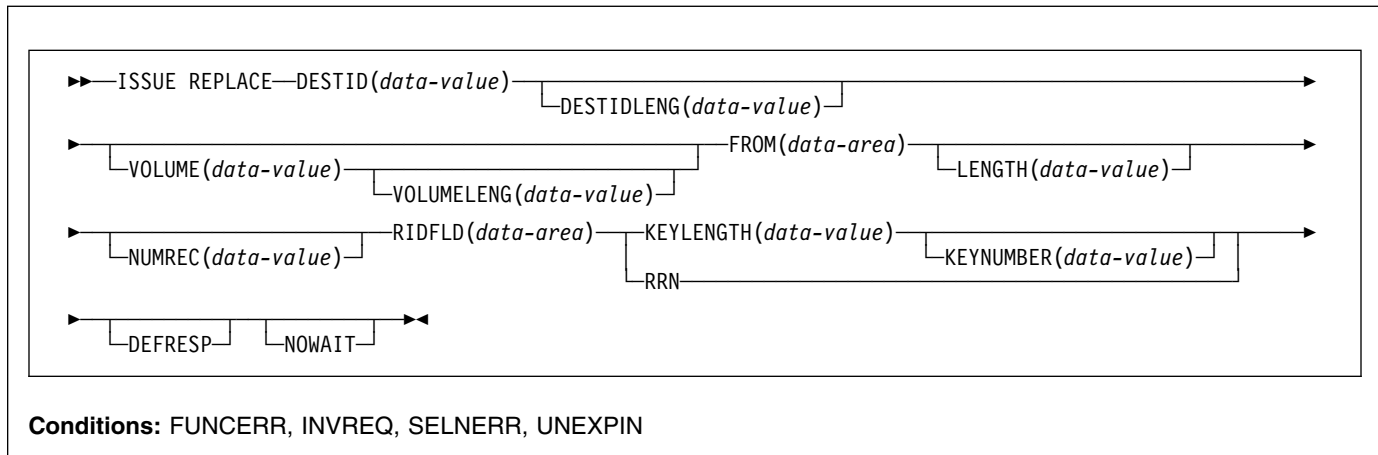
UNEXPIN

occurs when some unexpected or unrecognized information is received from the outboard controller.

Default action: terminate the task abnormally.

ISSUE REPLACE

Update a record in a data set.



Description

ISSUE REPLACE updates (replaces) a record in either a relative (addressed direct) or an indexed (keyed direct) data set in an outboard controller.

Options

DEFRESP

specifies that all terminal control commands issued as a result of the ISSUE REPLACE command request a definite response from the outboard batch program, irrespective of the specification of message integrity for the CICS task (by the system programmer).

DESTID(*data-value*)

specifies the name (1–8 characters) of the data set in the outboard destination.

DESTIDLENG(*data-value*)

specifies the length (halfword binary value) of the name specified in the DESTID option.

FROM(*data-area*)

specifies the data that is to be written to the data set.

KEYLENGTH(*data-value*)

specifies the length (halfword binary value) of the key specified in the RIDFLD option.

KEYNUMBER(*data-value*)

specifies the number, as a halfword binary value, of the index to be used to locate the record. There can be eight indexes (1 through 8). The default is 1. This option applies only to DPCX/DXAM and is mutually exclusive with RRN.

LENGTH(*data-value*)

specifies the length (halfword binary value) of the data to be written.

NOWAIT

specifies that the CICS task continues processing without waiting for the ISSUE REPLACE command to complete. If this option is not specified, the task activity is suspended until the command is completed.

NUMREC(*data-value*)

for a relative data set, specifies as a halfword binary value the number of logical records to be replaced. Records are replaced sequentially starting with the one identified by the RIDFLD option.

For an indexed data set, NUMREC cannot be specified because only one record is replaced.

RIDFLD(*data-area*)

specifies the record identification field.

For a relative record data set, the RIDFLD option specifies a fullword binary integer (the relative record number of the record starting from zero); and the RRN option is used.

For an indexed data set, the RIDFLD option specifies the key that is embedded in the data specified by the FROM option. The KEYLENGTH option is also required.

RRN

specifies that the record identification field specified in the RIDFLD option contains a relative record number. This option is required for a relative record data set.

If the option is not specified, RIDFLD is assumed to specify a key.

VOLUME(*data-value*)

specifies the name (1–6 characters) of a diskette in an outboard destination that contains the data set specified in the DESTID option.

VOLUMELENG(*data-value*)

specifies the length (halfword binary value) of the name specified in the VOLUME option.

Conditions

FUNCERR

occurs if there is an error during execution of the command. Destination selection is unaffected and other commands for the same destination may be successful.

Default action: terminate the task abnormally.

INVREQ

RESP2 values:

200 A distributed program link server application specified the function-shipping session (its principal facility) on the CONVID option.

Default action: terminate the task abnormally.

SELNERR

occurs if there is an error during destination selection. The destination is not selected and other commands for the same destination are unlikely to be successful.

Default action: terminate the task abnormally.

UNEXPIN

occurs when some unexpected or unrecognized information is received from the outboard controller.

Default action: terminate the task abnormally.

ISSUE RESET

ISSUE RESET

Relinquish use of a telecommunication line.

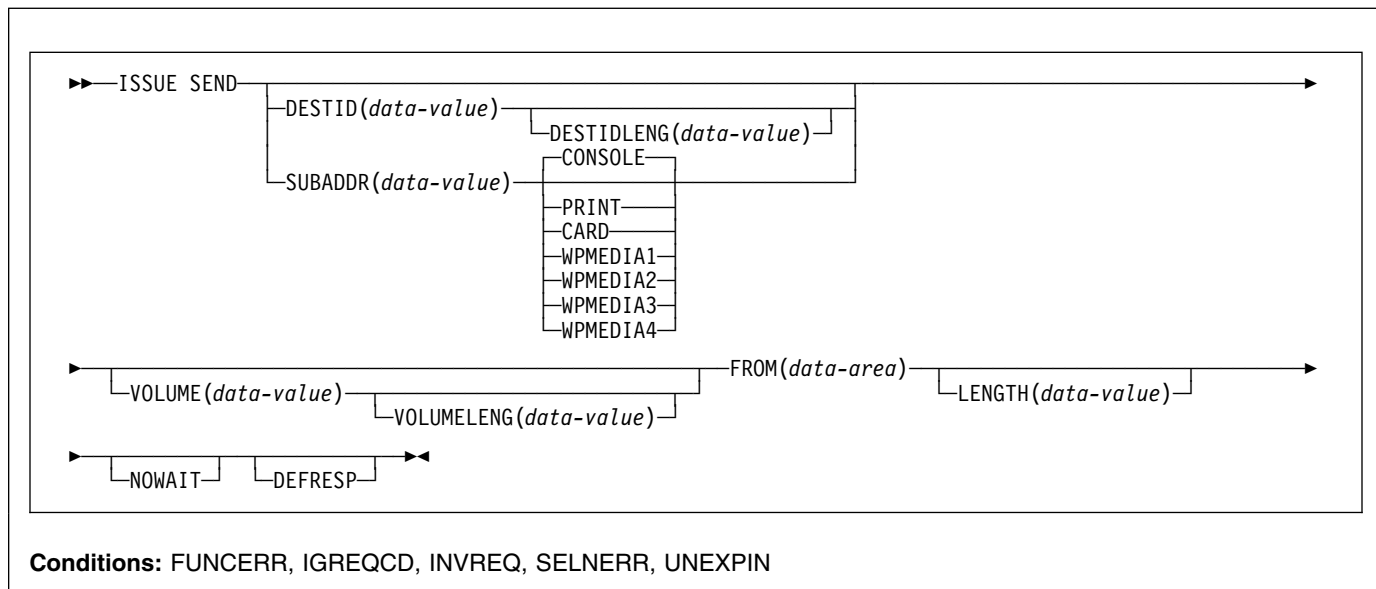
```
▶▶—ISSUE RESET—◀◀
```

Condition: INVREQ, NOTALLOC

This command is retained for compatibility only. You should be using ISSUE DISCONNECT.

ISSUE SEND

Send data to a named data set or to a selected medium.



Description

ISSUE SEND sends data to a named data set in an outboard controller, or to a selected medium in a batch logical unit or an LUTYPE4 logical unit. The options CONSOLE, PRINT, CARD, and WPMEDIA1–4 are alternatives to DESTID and DESTIDLENG.

Options

CARD

specifies that the output medium is a card reader or card punch device. This option is not valid with DESTID and DESTIDLENG.

CONSOLE

specifies that the output medium is that provided for messages to the operator. This option is not valid with DESTID and DESTIDLENG. This refers to a programmable subsystem such as the IBM 3790 data communication system. It does not refer to a CICS or system console.

DEFRESP

specifies that all terminal control commands issued as a result of the ISSUE SEND command request a definite response from the outboard batch program, irrespective of the specification of message integrity for the CICS task (by the system programmer).

DESTID(*data-value*)

specifies the name (1–8 characters) of the data set in the outboard destination.

DESTIDLENG(*data-value*)

specifies the length (halfword binary value) of the name specified in the DESTID option.

FROM(*data-area*)

specifies the data to be written to the data set.

LENGTH(*data-value*)

specifies a halfword binary value that is the length of the data to be written.

NOWAIT

specifies that the CICS task continues processing without waiting for the ISSUE SEND command to complete. If this option is not specified, the task activity is suspended until the command is completed.

PRINT

specifies that the output is to the print medium.

SUBADDR(*data-value*)

specifies the medium subaddress as a halfword binary value (in the range 0–15) that allows media of the same type, for example, “printer 1” or “printer 2”, to be defined. Value 15 means a medium of any type. The default is zero.

VOLUME(*data-value*)

specifies the name (1–6 characters) of a diskette in an outboard destination that contains the data set specified in the DESTID option.

ISSUE SEND

VOLUMELENG(*data-value*)

specifies the length of the name specified in the VOLUME option as a halfword binary value.

WPMEDIA1 through WPMEDIA4

specifies that for each specific LUTYPE4 device, a word processing medium is defined to relate to a specific input/output device.

Conditions

FUNCERR

occurs if there is an error during execution of the command. Destination selection is unaffected and other commands for the same destination may be successful.

Default action: terminate the task abnormally.

IGREQCD

occurs when an attempt is made to execute an ISSUE SEND command after a SIGNAL RCD data-flow control code has been received from an LUTYPE4 logical unit.

Default action: terminate the task abnormally.

INVREQ

RESP2 values:

200 A distributed program link server application specified the function-shipping session (its principal facility) on the CONVID option.

Default action: terminate the task abnormally.

SELNERR

occurs if there is an error during destination selection. The destination is not selected and other commands for the same destination are unlikely to be successful.

Default action: terminate the task abnormally.

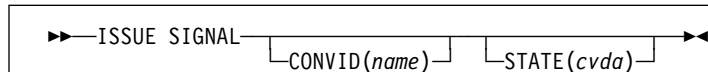
UNEXPIN

occurs when some unexpected or unrecognized information is received from the outboard controller.

Default action: terminate the task abnormally.

ISSUE SIGNAL (APPC)

Request change of direction from sending transaction on an APPC mapped conversation.



Conditions: INVREQ, NOTALLOC, TERMERR

Description

ISSUE SIGNAL, in a transaction in receive mode, signals to the sending transaction that a mode change is needed. It raises the SIGNAL condition on the next SEND, RECEIVE, or CONVERSE command executed in the sending transaction, and a previously executed HANDLE CONDITION command for this condition can be used either to take some action, or to ignore the request.

Options

CONVID(*name*)

identifies the conversation to which the command relates. The 4-character name identifies either the token returned by a previously executed ALLOCATE command in EIBRSRCE in the EIB, or the token representing the principal facility (returned by a previously executed ASSIGN command).

For compatibility with earlier releases, SESSION is accepted as a synonym for CONVID. New programs should use CONVID.

If both CONVID and SESSION are omitted, the principal facility is assumed.

STATE(*cvda*)

gets the state of the current conversation. The *cvda* values returned by CICS are:

ALLOCATED
 CONFFREE
 CONFRECEIVE
 CONFSEND
 FREE
 PENDFREE
 PENDRECEIVE
 RECEIVE
 ROLLBACK
 SEND
 SYNCFREE
 SYNCRECEIVE
 SYNCSEND

Conditions

INVREQ

RESP2 values:

200 A distributed program link server application specified the function-shipping session (its principal facility) on the CONVID option.

also occurs (RESP2 not set) in any of the following situations:

Default action: terminate the task abnormally.

NOTALLOC

occurs if the specified CONVID value does not relate to a conversation that is owned by the application.

Default action: terminate the task abnormally.

TERMERR

occurs for a session-related error. Any action on that conversation other than a FREE causes an ATCV abend.

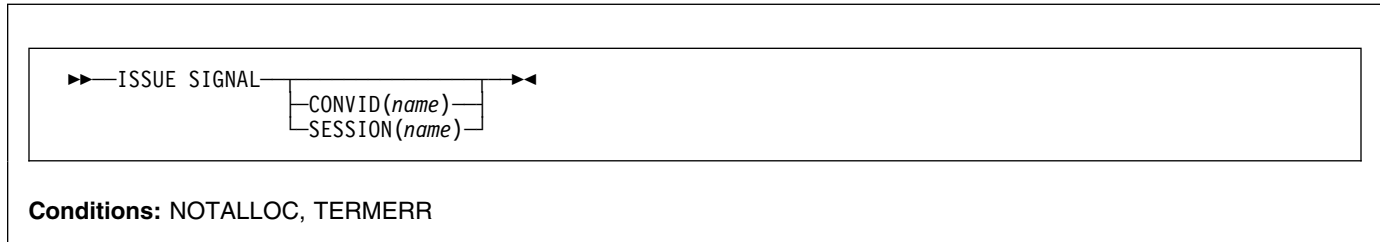
A CANCEL TASK request by a user node error program (NEP) may cause TERMERR if the task has an outstanding terminal control request when the node abnormal condition program handles the session error.

Default action: terminate the task abnormally with abend code ATNI.

ISSUE SIGNAL (LUTYPE6.1)

ISSUE SIGNAL (LUTYPE6.1)

Request change of direction from sending transaction on LUTYPE6.1 conversation.



Description

ISSUE SIGNAL, in a transaction in receive mode, signals to the sending transaction that a mode change is needed. It raises the SIGNAL condition on the next SEND, RECEIVE, or CONVERSE command executed in the sending transaction, and a previously executed HANDLE CONDITION command for this condition can be used either to take some action, or to ignore the request.

If both CONVID and SESSION are omitted, the principal facility for the task is used.

Options

CONVID(*name*)

identifies the conversation to which the command relates. The 4-character name identifies either the token returned by a previously executed ALLOCATE command in EIBRSRCE in the EIB, or the token representing the principal facility (returned by a previously executed ASSIGN command).

SESSION(*name*)

specifies the symbolic identifier (1–4 characters) of a session TCTTE. This option specifies the alternate facility to be used.

Conditions

NOTALLOC

occurs if the facility specified in the command is not owned by the application.

Default action: terminate the task abnormally.

TERMERR

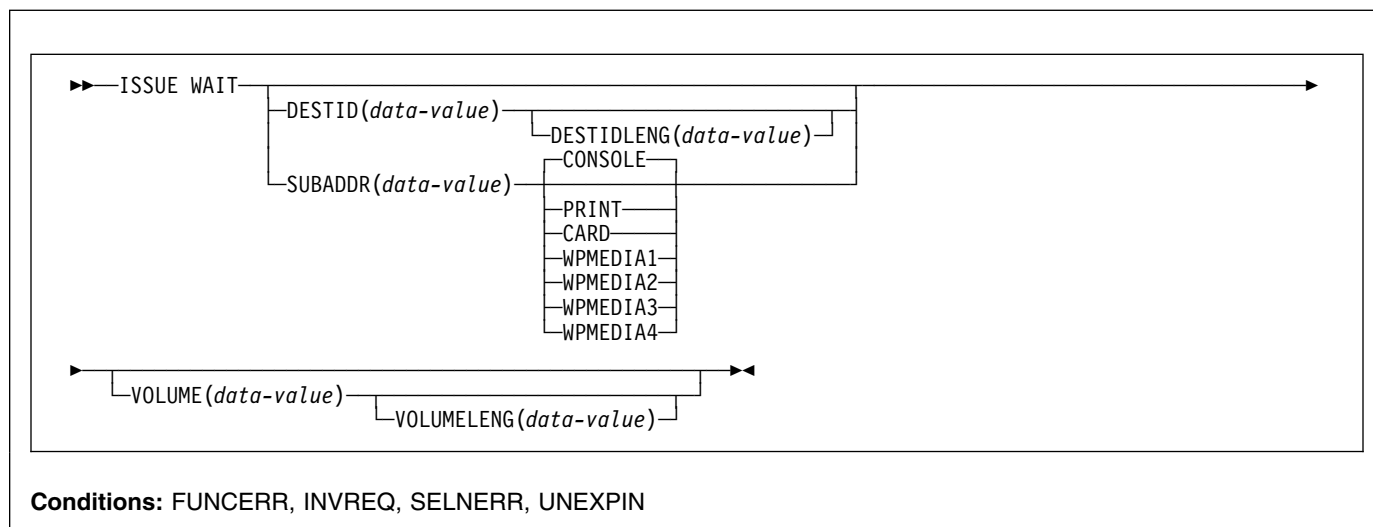
occurs for a session-related error. Any action on that conversation other than a FREE causes an ATCV abend.

A CANCEL TASK request by a user node error program (NEP) may cause TERMERR if the task has an outstanding terminal control request when the node abnormal condition program handles the session error.

Default action: terminate the task abnormally with abend code ATNI.

ISSUE WAIT

Wait for an operation to be completed.



Description

ISSUE WAIT suspends task activity until the previous batch data interchange command is completed. This command is meaningful only when it follows an ISSUE ADD, ISSUE ERASE, ISSUE REPLACE, or ISSUE SEND command. The options CONSOLE, PRINT, CARD, and WPMEDIA1–4 are alternatives to DESTID and DESTIDLENG.

Options

CARD

specifies that the output medium is a card reader or card punch device. This option is not valid with DESTID and DESTIDLENG.

CONSOLE

specifies that the output medium is that provided for messages to the operator. This option is not valid with DESTID and DESTIDLENG.

This refers to a programmable subsystem such as the IBM 3790 data communication system. It does not refer to a CICS or system console.

DESTID(*data-value*)

specifies the name (1–8 characters) of the data set in the outboard destination.

DESTIDLENG(*data-value*)

specifies the length (halfword binary value) of the name specified in the DESTID option.

PRINT

specifies that the output is to the print medium.

SUBADDR(*data-value*)

specifies the medium subaddress as a halfword binary value (in the range 0–through 15) that allows media of the same type, for example, “printer 1” or “printer 2”, to be defined. Value 15 means a medium of any type. The default is zero.

VOLUME(*data-value*)

specifies the name (1–6 characters) of a diskette in an outboard destination that contains the data set specified in the DESTID option.

VOLUMELENG(*data-value*)

specifies the length of the name specified in the VOLUME option as a halfword binary value.

WPMEDIA1 through WPMEDIA4

specifies that, for each specific LUTYPE4 device, a word-processing medium is defined to relate to a specific input/output device.

Conditions

FUNCERR

occurs if there is an error during execution of the command. Destination selection is unaffected and other commands for the same destination may be successful.

Default action: terminate the task abnormally.

INVREQ

RESP2 values:

200 A distributed program link server application specified the function-shipping session (its principal facility) on the CONVID option.

ISSUE WAIT

Default action: terminate the task abnormally.

SELNERR

occurs if there is an error during destination selection. The destination is not selected and other commands for the same destination are unlikely to be successful.

Default action: terminate the task abnormally.

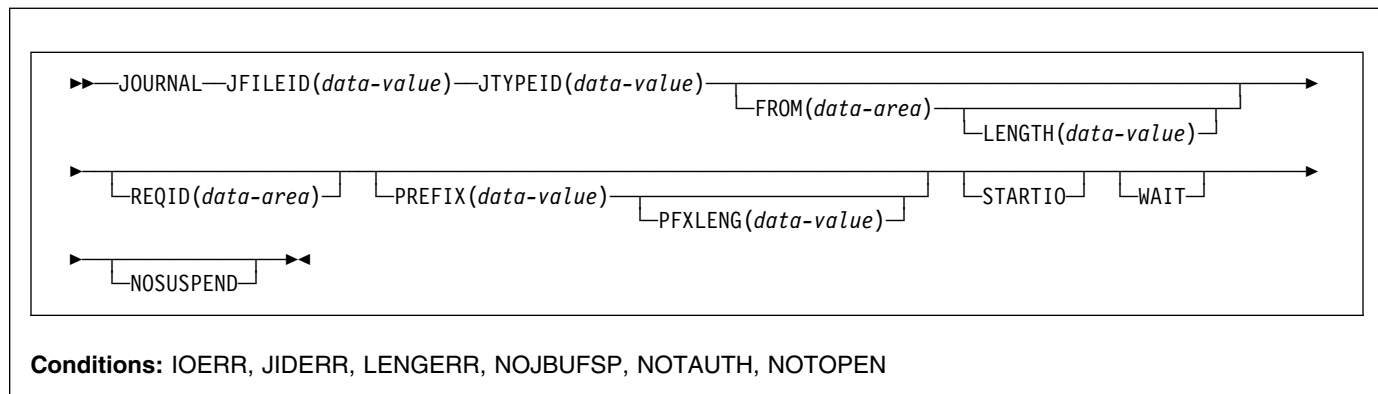
UNEXPIN

occurs when some unexpected or unrecognized information is received from the outboard controller.

Default action: terminate the task abnormally.

JOURNAL

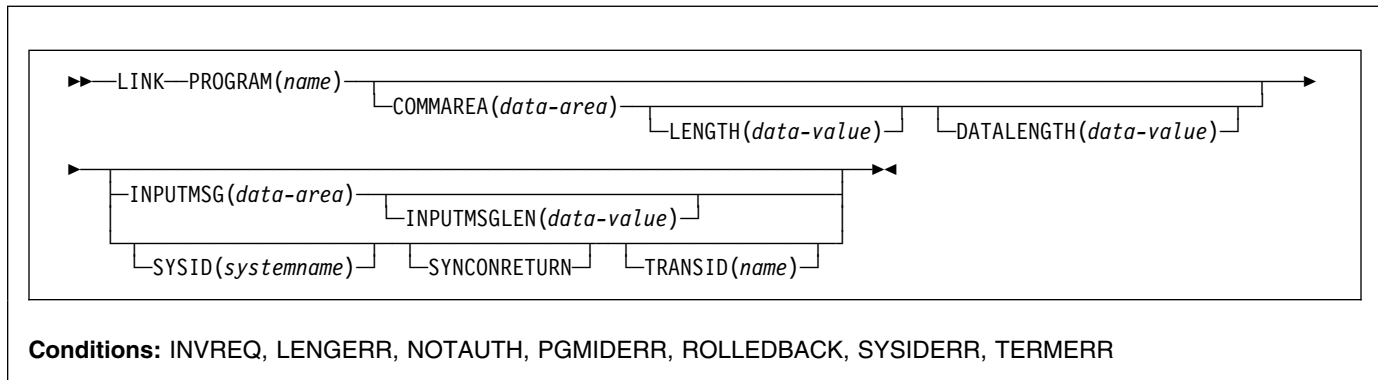
Create a journal record.



This command is now superseded by WAIT JOURNALNUM, but is still supported for compatibility with previous releases.

LINK

Link to another program expecting return.



Description

LINK passes control from an application program at one logical level to an application program at the next lower logical level. If the requested program is not defined to CICS, and AUTOINSTALL is active, CICS supplies a definition for the program. If it is not a remote program in another CICS region, and the linked-to program is not already in main storage, CICS loads it. If the SYSID option specifies the name of a remote CICS region, CICS ships the link request to the remote region. When the RETURN command is executed in the linked-to program, control is returned to the program initiating the link at the next sequential executable instruction.

The linked-to program operates independently of the program that issues the LINK command with regard to handling conditions, attention identifiers, abends, and execution key. For example, the effects of HANDLE CONDITION commands in the linking program are not inherited by the linked-to program, but the original HANDLE CONDITION commands are restored on return to the linking program. See the *CICS Application Programming Guide* for more information and an illustration of the concept of logical levels.

You can use the HANDLE ABEND command to deal with abnormal terminations in other link levels. See the *CICS Application Programming Guide* for further details about the relationship between LINK and HANDLE ABEND.

Note: The external CICS interface (EXCI) provides a LINK command that performs all six commands of the interface in one invocation.

Distributed program link

If you specify a remote region name on the SYSID option (with or without the associated TRANSID and SYNCONRETURN options), the link is a *distributed program link* (DPL). In response to a distributed program link, the local CICS region (the *client region*) ships the link request to the remote region (the *server region*). The server region executes the linked-to program (the *server program*) on behalf of the program issuing the link request (the *client program*).

The SYSID and INPUTMSG options are mutually exclusive. If you specify both options on a LINK command, the translator issues error message DFH7230 (severity E) indicating conflicting options. See the DFH7xxx (DFHEXP command translator diagnostic) messages entry in the *VSE/ESA Messages and Codes Volume 3* manual for an explanation of severity E for the different supported languages.

A server program running in the server region is restricted to a DPL subset of the CICS API. Briefly, the server program cannot issue:

- Terminal control commands that reference the principal facility
- Options of ASSIGN that return terminal attributes
- BMS commands
- Signon and signoff commands
- Batch data interchange commands
- Commands that address the TCTUA

For details of the restricted DPL subset of the API, see Appendix G, "API restrictions for distributed program link" on page 397.

Abends in the server program: If a server program abends, the abend code is returned to the client program. If the client program is not written to handle the abend returned

by the server program, the client program abends with the same abend code returned by the server program.

You cannot use DPL to link to the CICS master terminal program, DFHEMTA, or to the RDO program, DFHEDAP. The addresses passed as parameters to DFHEMTA and DFHEDAP are valid only in the region that issues the EXEC CICS LINK command, which means you cannot route a DFHEMTA or DFHEDAP request to a remote CICS.

Options

COMMAREA(*data-area*)

specifies a communication area that is to be made available to the invoked program. In this option a pointer to the data area is passed. In a COBOL receiving program, you must give this data area the name DFHCOMMAREA. (See the section about passing data to other programs in the *CICS Application Programming Guide*.)

DATALENGTH(*data-value*)

specifies a halfword binary value that is the length of a contiguous area of storage, from the start of the COMMAREA, to be passed to the invoked program. If the amount of data being passed in a COMMAREA is small, but the COMMAREA itself is large so that the linked-to program can return the requested data, you should specify DATALENGTH in the interest of performance.

DATALENGTH cannot be used at the same time as INPUTMSG.

INPUTMSG(*data-area*)

specifies data to be supplied to the invoked program when it first issues a RECEIVE command. This data remains available until the execution of a RECEIVE or RETURN command. An invoked program can invoke a further program and so on, creating a chain of linked programs. If a linked-to chain exists, CICS supplies the INPUTMSG data to the first RECEIVE command executed in the chain. If control returns to the program that issued the LINK with INPUTMSG before the INPUTMSG data has been accepted by a RECEIVE command, CICS assumes that a RECEIVE command has been issued. This means that the original INPUTMSG data is no longer available.

INPUTMSG cannot be used at the same time as DATALENGTH.

See also the *CICS Application Programming Guide* for more information about the INPUTMSG option.

INPUTMSGLEN(*data-value*)

specifies a halfword binary value to be used with INPUTMSG.

LENGTH(*data-value*)

specifies a halfword binary value that is the length in bytes of the COMMAREA (communication area). For a

description of a safe upper limit, see “LENGTH options” on page 5.

PROGRAM(*name*)

specifies the identifier (1–8 characters) of the program to which control is to be passed unconditionally. The specified name must have been defined as a program to CICS, though if AUTOINSTALL is active a definition is autoinstalled. If the SYSID option specifies a remote region, the linked-to program is the server program in the server region.

Note the use of quotes:

```
EXEC CICS LINK PROGRAM('PROGX')
```

PROGX is in quotes because it is the program name.

```
EXEC CICS LINK PROGRAM(DAREA)
```

DAREA is not in quotes because it is the name of a data area that contains the actual program name.

SYNCONRETURN

specifies that the server region named on the SYSID option is to take a syncpoint on successful completion of the server program.

Synconreturn is only applicable to remote links, it is ignored if the link is local.

SYSID(*systemname*)

specifies the system name of a CICS server region to where the program link request is to be routed. Although this option is primarily for a distributed program link, you can specify the name of the local CICS region, in which case CICS treats the LINK request as a normal local link request.

The SYSID specified on the LINK command takes priority over any remote system specified on the RDO PROGRAM resource definition.

TRANSID(*name*)

specifies the name of the mirror transaction that the remote region is to attach, and under which it is to run the server program. If you omit the TRANSID option, the server region attaches either CSMI, CPMI, or CVMI by default.

The transaction name you specify on the LINK command takes priority over any transaction specified on the program resource definition. Whilst you can specify your own name for the mirror transaction initiated by DPL requests, the transaction *must* be defined in the server region, and the transaction definition must specify the mirror program, DFHMIRS.

LINK

Conditions

INVREQ

RESP2 values:

- 8 A LINK command with the INPUTMSG option is issued for a program that is not associated with a terminal, or that is associated with an APPC logical unit, or an IRC session.
- 14 The SYNCONRETURN option is specified but the program issuing the link request (the client program) is already in conversation with a mirror task (that is, a unit of work (LUW) is in progress) in the remote region specified on the SYSID option. In this case, the client program is in an incorrect state to support the SYNCONRETURN option.
- 15 The program issuing the link request is already in conversation with a mirror task and the TRANSID specified is different from the transaction identifier of the active mirror.
- 16 The TRANSID specified is all blanks.
- 19 A LINK command with the INPUTMSG option is issued for a program that is the subject of a DPL request; that is, SYSID is also specified.
- 30 The program manager domain has not yet been initialized. This is probably due to a link request having been made in a first stage PLT.

Default action: terminate the task abnormally.

Note: RESP2 values are not returned to the client for conditions occurring in a DPL server program.

LENGERR

RESP2 values:

- 11 The COMMAREA length is less than 0 or greater than 32767.
- 12 The length specified on the DATALENGTH option is a negative value.
- 13 The length specified on the DATALENGTH option is greater than the length specified on the LENGTH option.
- 26 The COMMAREA address is zero, but the COMMAREA length is non zero.
- 27 The INPUTMSG length is less than 0 or greater than 32767.
- 28 The length specified on the LENGTH option is greater than the length of the data area specified in the COMMAREA option, and while that data was being copied a destructive overlap occurred because of the incorrect length.

Default action: terminate the task abnormally.

Note: RESP2 values are not returned to the client for conditions occurring in a DPL server program.

NOTAUTH

RESP2 values:

- 101 A resource security check has failed on PROGRAM(name).

Default action: terminate the task abnormally.

PGMIDERR

RESP2 values:

- 1 A program has no entry in the PPT and either program autoinstall was switched off, or the program autoinstall control program indicated that the program should not be autoinstalled.
- 2 A program is disabled.
- 3 A program could not be loaded because:
 - This was the first load of the program and the program load failed, usually because the load module could not be found.
 - This was a subsequent load of the program, but the first load failed.

In order to reset the load status the load module must be in the LIBDEF search chain of the CICS startup job stream and a SET PROGRAM NEWCOPY will be required.

- 21 The program autoinstall control program failed either because the program autoinstall control program is incorrect, incorrectly defined, or as a result of an abend in the program autoinstall control program. Program autoinstall is disabled and message DFHPG0202 or DFHPG0203 written to the CSPL.
- 22 The model returned by the program autoinstall control program was not defined in the PPT table, or was not enabled.
- 23 The program autoinstall control program returned invalid data.
- 24 Define for the program failed due to autoinstall returning an invalid program name or definition.

Default action: terminate the task abnormally.

ROLLEDBACK

RESP2 values:

- 29 The SYNCONRETURN is specified and the server program is unable successfully to take a syncpoint. The server program has taken a rollback, and all changes made to recoverable resources in the remote region, within the current unit of work, are backed out.

Default action: terminate the task abnormally.

SYSIDERR

RESP2 values:

- 18** The SYSID specified cannot be found in the intersystem table, or the link to the specified system is unavailable.
- 20** The remote system specified by SYSID is an LUTYPE6.1-connected system. Distributed program link requests are not supported on LUTYPE6.1 connections.

also occurs (RESP2 not set) on DPL requests where the remote (or daisy-chained) system has not been initialized or requested resources are not defined to the remote CICS or related system components.

Note: There is no local queuing in the event of a SYSIDERR.

Default action: terminate the task abnormally.

TERMERR

RESP2 values:

- 17** An irrecoverable error occurs during the conversation with the mirror (for example, if the session fails, or if the server region fails).

Default action: terminate the task abnormally.

Note: RESP2 values are not returned to the client for conditions occurring in a DPL server program.

Examples

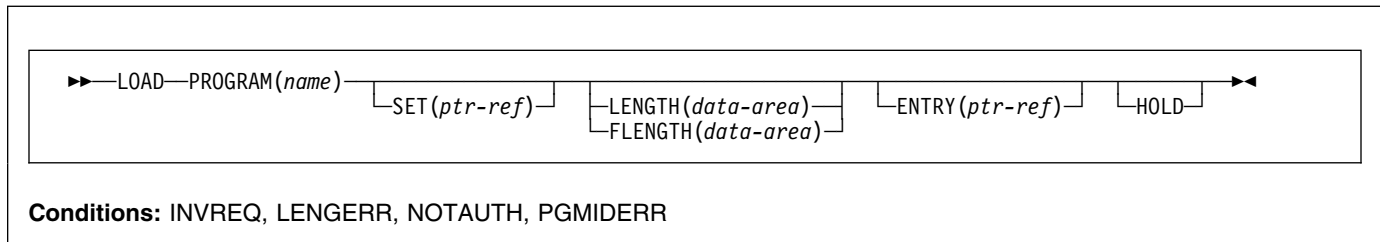
The following example shows how to request a link to an application program called PROGRAM:

```
EXEC CICS LINK PROGRAM(PROGNAME)
          COMMAREA(COMA) LENGTH(LENA)
          DATALENGTH(LENI) SYSID('CONX')
```

LOAD

LOAD

Load a program from a LIBDEF PHASE SEARCH sublibrary into main storage.



Conditions: INVREQ, LENGERR, NOTAUTH, PGMIDERR

Dynamic transaction routing

Using LOAD with HOLD, or using a PROGRAM resource that has been defined with RELOAD(YES) in its RDO PROGRAM resource definition, could create inter-transaction affinities that adversely affect the use of dynamic transaction routing. See the *CICS Application Programming Guide* for more information about transaction affinities.

Description

Load makes available to the invoking task a copy of an application program or map. If the program is defined with RELOAD(NO) in its RDO PROGRAM resource definition, it is only fetched from the library where it resides if there is not a copy already in main storage. If the program is defined with RELOAD(YES), a new copy is always fetched from the library. (See the *CICS Application Programming Guide* for further details about maps.) Using LOAD can reduce system overhead.

Options

ENTRY(ptr-ref)

specifies the pointer reference that is to be set to the address of the entry point in the program that has been loaded.

The top bit of the address is set on if the program is defined with AMODE(31).

For assembler programs without an explicit ENTRY defined in the linkedit definitions, the entry point returned depends on (1) whether there is a CICS stub, and (2) whether the LOAD command is issued from a PLT program:

- If there is a CICS stub, the entry point address is incremented for this stub unless the LOAD command is issued from a PLT program executed during the first phase of initialization or the final phase of shutdown.
- If there is not a CICS stub, the entry point address is the same as the load point address.

FLENGTH(data-area)

specifies a fullword binary area to be set to the length of the loaded program, table, or map. Use FLENGTH if the length of the loaded program is greater than 32KB.

HOLD

specifies that the loaded program, table, or map is not to be released (if still available) when the task issuing the LOAD command is terminated; it is to be released only in response to a RELEASE command from this task or from another task.

If you omit HOLD, the program, table, or map is released when the task that issued the load terminates or issues a RELEASE command.

If, however, the program is defined with RELOAD(YES), neither of the above apply. RELEASE does not work, and a FREEMAIN must be issued to get rid of the program.

LENGTH(data-area)

specifies a halfword binary value to be set to the length of the loaded program, table, or map. To avoid raising the LENGERR condition, use FLENGTH if the length of the loaded program is likely to be greater than 32KB.

PROGRAM(name)

specifies the identifier (1–8 characters) of a program, table, or map to be loaded. The specified name must have been defined as a program to CICS, though if AUTOINSTALL is active a definition is autoinstalled.

SET(ptr-ref)

specifies the pointer reference that is to be set to the address at which a program, table, or map is loaded.

Conditions

INVREQ

RESP2 values:

- 30** The program manager domain has not yet been initialized. This is probably due to a load request having been made in a first stage PLT.

Default action: terminate the task abnormally.

LENGERR

RESP2 values:

- 19** LENGTH is used and the length of the loaded program is not less than 32KB.

Default action: terminate the task abnormally.

NOTAUTH

RESP2 values:

- 101** A resource security check has failed on PROGRAM(name).

Default action: terminate the task abnormally.

PGMIDERR

RESP2 values:

- 1** A program, table, or map has no entry in the PPT and either program autoinstall was switched off, or the program autoinstall control program indicated that the program should not be autoinstalled.
- 2** A program is disabled.
- 3** A program could not be loaded because:
- This was the first load of the program and the program load failed, usually because the load module could not be found.
 - This was a subsequent load of the program, but the first load failed.

In order to reset the load status the load module must be in the LIBDEF search chain of the CICS startup job stream, and a SET PROGRAM NEWCOPY will be required.

- 9** The installed program definition is for a remote program.
- 21** The program autoinstall control program failed either because the program autoinstall control program is incorrect, incorrectly defined, or as a result of an abend in the program autoinstall control program. Program autoinstall is disabled and message DFHPG0202 or DFHPG0203 written to the CSPL.
- 22** The model returned by the program autoinstall control program was not defined in the PPT table, or was not enabled.
- 23** The program autoinstall control program returned invalid data.

- 24** Define for the program failed due to autoinstall returning an invalid program name or definition.

Default action: terminate the task abnormally.

Examples

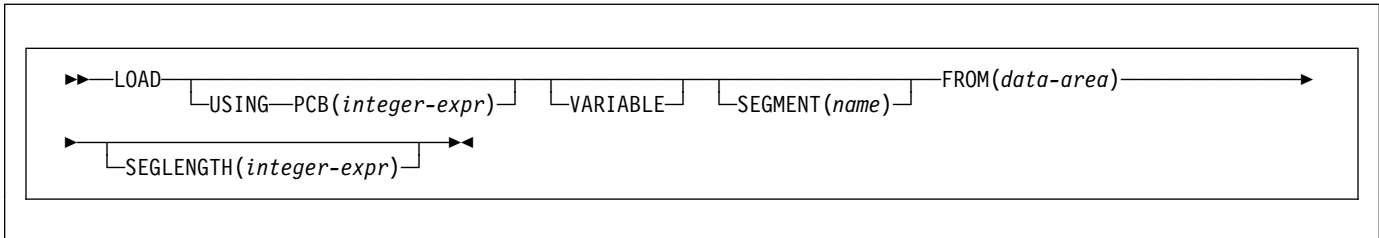
The following example shows how to load a user-prepared table called TB1:

```
EXEC CICS LOAD PROGRAM('TB1') SET(PTR)
```

LOAD (DL/I)

LOAD (DL/I)

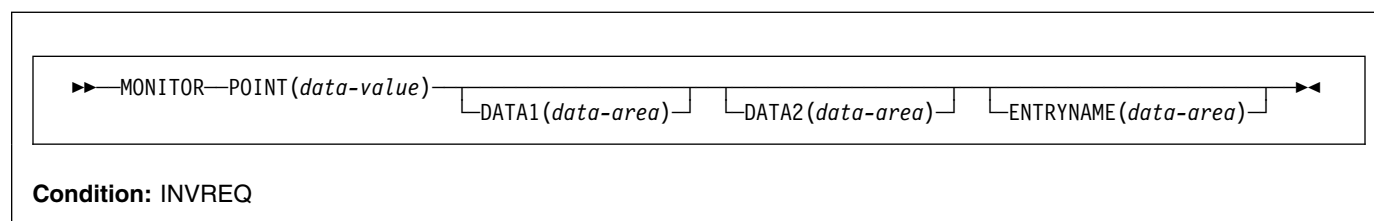
Load a segment (batch only).



See *DL/I DOS/VS Application Programming: High Level Programming Interfaces* for details.

MONITOR

Code a user event-monitoring point.



Description

MONITOR provides information about the performance of your application transactions. It replaces the monitoring aspects of ENTER TRACEID.

In addition to the monitoring data collected at predefined event monitoring points (EMPs) within CICS, a user application program can contribute data to user fields within the CICS monitoring records. You can do this by using the MONITOR command to invoke user-defined EMPs. At each of these user EMPs, you can add or change 1–16384 bytes of your own data in each performance monitoring record. In those 16384 bytes, you can have any combination of the following:

- 0 through 256 counters
- 0 through 256 clocks
- A single 8192-byte character string

Options

DATA1(*data-area*)

specifies a 4-byte variable whose contents depend on the type of user EMP being used:

- If the user EMP contains an ADDCNT, SUBCNT, NACNT, EXCNT, or ORCNT option, the DATA1 variable is an area used as defined by the MCT user EMP definition.
- If the MCT user EMP definition contains an MLTCNT option, the DATA1 variable is an area with the address of a series of adjacent fullwords containing the values to be added to the user count fields defined in the MCT user EMP definition.
- If the MCT user EMP definition contains a MOVE option, the DATA1 variable is an area with the address of the character string to be moved.

See the *CICS Resource Definition Guide* for details of user EMP options.

DATA2(*data-area*)

specifies a 4-byte variable whose contents depend on the type of user EMP being used:

- If the user EMP contains an ADDCNT, SUBCNT, NACNT, EXCNT, or ORCNT option, the DATA2 variable is an area used as defined by the MCT user EMP definition.
- If the MCT user EMP definition contains an MLTCNT option, the DATA2 variable is an area with the number of user count fields to be updated. The number specified in DATA2 overrides the default value defined in the MCT for the operation. A value of zero instructs monitoring to use the default. If DATA2 is not specified, the MLTCNT operation raises an INVREQ condition although the operation was successful.
- If the MCT user EMP definition contains a MOVE option, the DATA2 variable is an area with the length of the character string to be moved. The number specified in DATA2 will override the default value defined in the MCT for the operation. A value of zero instructs monitoring to use the default. If DATA2 is not specified, the MOVE operation raises an INVREQ although the operation was successful.

See the *CICS Resource Definition Guide* for details of user EMP options.

ENTRYNAME(*data-area*)

is the monitoring point entry name that qualifies the POINT value and is defined in the monitoring control table (MCT). ENTRYNAME defaults to USER if not specified.

POINT(*data-value*)

specifies the monitoring point identifier as defined in the MCT, and is in the range 0 through 255. Note, however, that point identifiers in the range 200 through 255 are reserved for use by IBM program products.

Conditions

INVREQ

RESP2 values:

- 1 Your POINT value is outside the range 1 through 255.

MONITOR

- 2 Your POINT value is not defined in the MCT.
- 3 Your DATA1 value is not valid.
- 4 Your DATA2 value is not valid.
- 5 You did not specify DATA1 for an MCT operation that required it.
- 6 You did not specify DATA2 for an MCT operation that required it.

Default action: terminate the task abnormally.

Examples

For example, you could use these user EMPs to count the number of times a certain event occurs, or to time the interval between two events.

Figure 1 gives examples of MONITOR commands (and of the MCT entries you need for them). See the *CICS Customization Guide* for more information about monitoring.

Notes:

1. Example 1 shows a user clock being started by an application identified as PROG3. This is the eleventh EMP in this application. To prevent confusion with the eleventh EMP in another application, this EMP is uniquely identified by the tag ENTRY3.11. The clock that is being started is the first clock in a string.
2. Example 2 shows the same user clock being stopped, by the same application, but from a different EMP. The EMP is uniquely identified by the tag ENTRY3.12.
3. Example 3 shows some user data being loaded into the 32-byte character string reserved for that purpose. The loading starts at offset 0, and the data is no more than 32 bytes in length.

```
1:
EXEC CICS MONITOR
      POINT(11)
      ENTRYNAME(ENTRY3)
      needing: DFHMCT TYPE=EMP,
              CLASS=PERFORM,
              ID=(ENTRY3.11),
              CLOCK=(1,CLOCKA),
              PERFORM=SCLOCK(1)

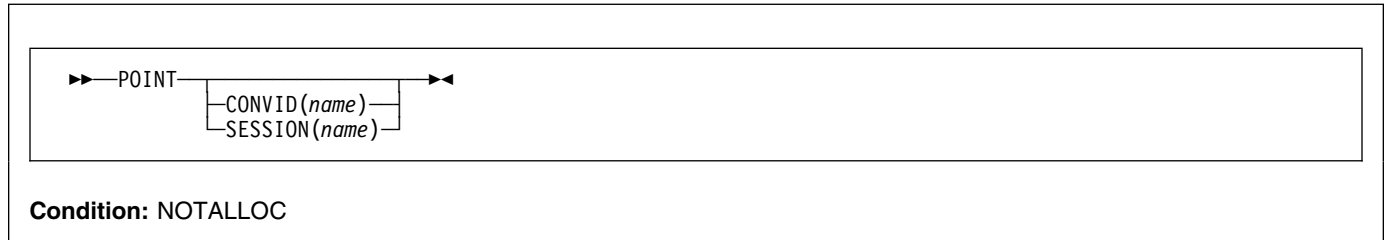
2:
EXEC CICS MONITOR
      POINT(12)
      ENTRYNAME(ENTRY3)
      needing: DFHMCT TYPE=EMP,
              CLASS=PERFORM,
              ID=(ENTRY3.12),
              PERFORM=PCLOCK(1)

3:
EXEC CICS MONITOR
      POINT(13)
      DATA1(address of data)
      DATA2(length of data)
      ENTRYNAME(ENTRY3)
      needing: DFHMCT TYPE=EMP,
              CLASS=PERFORM,
              ID=(ENTRY3.13),
              PERFORM=MOVE(0,32)
```

Figure 1. Examples of coding user EMPs

POINT

Get information about an LUTYPE6.1 logical unit.



Description

POINT gets information about a named facility, such as whether it owns the given facility.

This command can be used on an MRO session.

Options

CONVID(*name*)

identifies the conversation to which the command relates. The 4-character name identifies either the token returned by a previously executed ALLOCATE command in EIBRSRCE in the EIB, or the token representing the principal session (returned by a previously executed ASSIGN command).

SESSION(*name*)

specifies the symbolic identifier (1–4 characters) of a session TCTTE. This option specifies the alternate facility to be used. If both this option and CONVID are omitted, the principal facility for the task is used.

Conditions

NOTALLOC

occurs if the facility specified in the command is not owned by the application.

Default action: terminate the task abnormally.

POP HANDLE

POP HANDLE

Restore the stack.



```
▶▶—POP HANDLE—◀◀
```

Condition: INVREQ

Description

POP HANDLE enables you to restore the effect of IGNORE CONDITION, HANDLE ABEND, HANDLE AID, and HANDLE CONDITION commands to the state they were in before a PUSH HANDLE command was executed at the current link level. This can be useful, for example, during a branch to a subroutine embedded in a main program.

Normally, when a CICS program calls a subroutine (at the same logical level), the program or routine that receives control inherits the current HANDLE commands. These commands may not be appropriate within the called program. The called program can use PUSH HANDLE to suspend existing HANDLE commands, and before returning control to the caller, can then restore the original commands using the POP HANDLE command.

Note: When a CICS program uses EXEC CICS LINK to call another CICS program, the HANDLE effects are NOT inherited by the linked-to program, but CICS will search preceding logical levels for a HANDLE ABEND exit. See *CICS Application Programming Guide* for further details about the relationship between LINK and HANDLE ABEND.

You can nest PUSH HANDLE ... POP HANDLE command sequences within a task. Each POP HANDLE command restores a set of specifications.

The C language does not support POP HANDLE.

Conditions

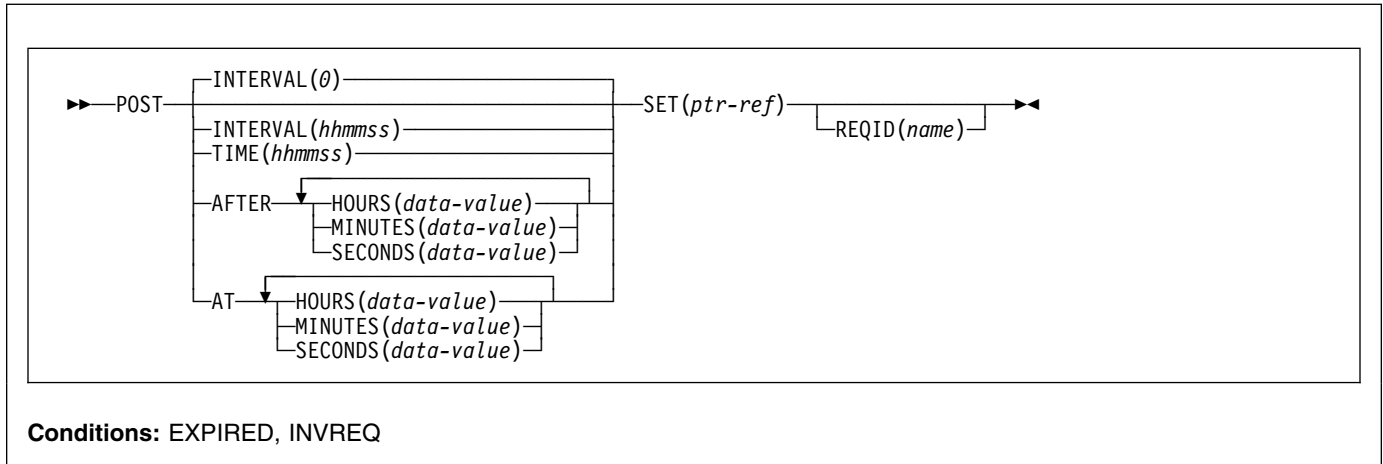
INVREQ

occurs if no matching PUSH HANDLE command has been executed at the current link level.

Default action: terminate the task abnormally.

POST

Request notification when a specified time has expired.



Note for dynamic transaction routing

Using POST if later CANCELED by another task could create inter-transaction affinities that adversely affect the use of dynamic transaction routing. See the *CICS Application Programming Guide* for more information about transaction affinities.

Description

POST requests notification that a specified time has expired. In response to this command, CICS makes a timer-event control area available for testing. This 4-byte control area is initialized to binary zeros, and the pointer reference specified in the SET option is set to its address.

When the time you specify has expired, the timer-event control area is posted; that is, its first byte is set to X'40' and its third byte to X'80'. You can test posting in either of the following ways:

- By checking the timer-event control area at intervals. You must give CICS the opportunity to post the area; that is, the task must relinquish control of CICS before you test the area. Normally, this condition is satisfied as a result of other commands being issued; if a task is performing a long internal function, you can force control to be relinquished by issuing a SUSPEND command.
- By suspending task activity by a WAIT EVENT or WAIT EXTERNAL command until the timer-event control area is posted. This action is similar to issuing a DELAY command but, with a POST and WAIT EVENT or WAIT EXTERNAL command sequence, you can do some processing after issuing the POST command; a DELAY command suspends task activity at once. No other task should attempt to wait on the event set up by a POST command.
- By using WAITCICS.

The timer-event control area can be released for a variety of reasons. If this happens, the result of any other task issuing a WAIT command on the event set up by the POST command is unpredictable.

However, other tasks can cancel the event if they have access to the REQID associated with the POST command. (See the CANCEL command and the description of the REQID option.) A timer-event control area provided for a task is not released or altered (except as described above) until one of the following events occurs:

- The task issues a subsequent DELAY, POST, or START command.
- The task issues a CANCEL command to cancel the POST command.
- The task is terminated, normally or abnormally.
- Any other task issues a CANCEL command for the event set up by the POST command.

A task can have only one POST command active at any given time. Any DELAY, POST, or START command supersedes a POST command previously issued by the task.

The default is INTERVAL(0), but for C the default is AFTER HOURS(0) MINUTES(0) SECONDS(0).

POST

Options

AFTER

specifies the interval of time to elapse.

There are two ways to enter the time under AFTER and AT.

1. A combination of at least two of HOURS(0–99), MINUTES(0–59), and SECONDS(0–59). HOURS(1) SECONDS(3) would mean one hour and three seconds (the minutes default to zero).
2. As one of HOURS(0–99), MINUTES(0–5999), or SECONDS(0–359999). HOURS(1) means one hour. MINUTES(62) means one hour and two minutes. SECONDS(3723) means one hour, two minutes, and three seconds.

AT specifies the time of expiring. For the ways to enter the time, see the AFTER option.

HOURS(*data-value*)

specifies a fullword binary value in the range 0–99. This is a suboption of the AFTER and AT options. For its use and meaning, see the AFTER option.

INTERVAL(*hhmmss*)

specifies an interval of time that is to elapse from the time at which the POST command is issued. The **mm** and **ss** are in the range 0–59. The time specified is added to the current clock time by CICS when the command is executed to calculate the expiration time.

This option is used to specify when the posting of the timer-event control area should occur.

When using the C language, you are recommended to use the AFTER/AT HOURS, MINUTES, and SECONDS options as C does not provide a packed decimal data type. You may use INTERVAL, but if the value specified is **not** an integer constant, the application is responsible for ensuring that the value passed to CICS is in packed decimal format.

MINUTES(*data-value*)

specifies a fullword binary value in the range 0–59, when HOURS or SECONDS are also specified, or 0–5999 when MINUTES is the only option specified. This is a suboption of the AFTER and AT options. For its use and meaning, see the AFTER option.

REQID(*name*)

specifies a name (1–8 characters), which should be unique, to identify the POST request. Using this option to specify an application-defined name is one way to enable another transaction to cancel the POST request.

If you do not specify your own REQID, CICS generates a unique request identifier for you in the EIBREQID field of the EXEC interface block. This, like your own REQID, can be used by another transaction to cancel the POST request.

To enable other tasks to cancel unexpired POST requests, you must make the request identifier

dynamically available. For example, storing it in a TS queue, whose name is known to other applications that may want to cancel the POST request, is one way you can pass a request identifier to other transactions.

SECONDS(*data-value*)

specifies a fullword binary value in the range 0–59, when HOURS or MINUTES are also specified, or 0–359999 when SECONDS is the only option specified. This is a suboption of the AFTER and AT options. For its use and meaning, see the AFTER option.

SET(*ptr-ref*)

specifies the pointer reference to be set to the address of the 4-byte timer-event control area generated by CICS. This area is initialized to binary zeros; on expiration of the specified time, the first byte is set to X'40', and the third byte to X'80'.

The timer-event control area always resides below the 16MB line in shared dynamic storage (SDSA).

TIME(*hhmmss*)

specifies the time when the posting of the timer-event control area should occur.

When using the C language, you are recommended to use the AFTER/AT HOURS, MINUTES, and SECONDS options as C does not provide a packed decimal data type. You may use TIME, but if the value specified is **not** an integer constant, the application is responsible for ensuring that the value passed to CICS is in packed decimal format. See the section about expiration times in the *CICS Application Programming Guide*.

Conditions

EXPIRED

occurs if the time specified has already expired when the command is issued.

Default action: ignore the condition.

INVREQ

RESP2 values:

- 4 Hours are out of range.
- 5 Minutes are out of range.
- 6 Seconds are out of range.

also occurs (RESP2 not set) in any of the following situations:

- The POST command is not valid for processing by CICS.

Default action: terminate the task abnormally.

Examples

The following example shows you how to request a timer-event control area for a task, to be posted after 30 seconds:


```
EXEC CICS POST  
  INTERVAL(30)  
  REQID('RBL3D')  
  SET(PREF)
```

The following example shows you how to ask to be notified when the specified time of day is reached. Because no request identifier is specified in the command, CICS automatically assigns one and returns it to the application program in the EIBREQID field in the EIB.

```
EXEC CICS POST  
  TIME(PACKTIME)  
  SET(PREF)
```

PURGE MESSAGE

PURGE MESSAGE

Discontinue building a BMS logical message.

▶▶—PURGE MESSAGE—◀◀

Conditions: Full BMS: INVREQ, TSIOERR

Description

PURGE MESSAGE discontinues the building of a BMS logical message. It deletes the current logical message, including any pages of device-dependent data stream already written to CICS temporary storage. The application program may then build a new logical message.

The portions of the logical message already built in main storage or in temporary storage are deleted.

See Appendix K, “BMS macro summary” on page 411 for map definition macros.

PURGE MESSAGE is only available on full-function BMS. For further information about BMS, see the *CICS Application Programming Guide*.

Conditions

INVREQ

RESP2 values:

200 The command was called in a distributed program link server program.

Default action: terminate the task abnormally.

TSIOERR

occurs if there is an irrecoverable temporary storage input/output error.

Default action: terminate the task abnormally.

PUSH HANDLE

Suspend the stack.



►►—PUSH HANDLE—◄◄

Description

PUSH HANDLE enables you to suspend the current effect of IGNORE CONDITION, HANDLE ABEND, HANDLE AID, and HANDLE CONDITION commands. This can be useful, for example, during a branch to a subroutine embedded in a main program.

Normally, when a CICS program calls a subroutine at the same logical level, the program or routine that receives control inherits the current HANDLE commands. These commands may not be appropriate within the called program. The called program can use PUSH HANDLE to suspend existing HANDLE commands.

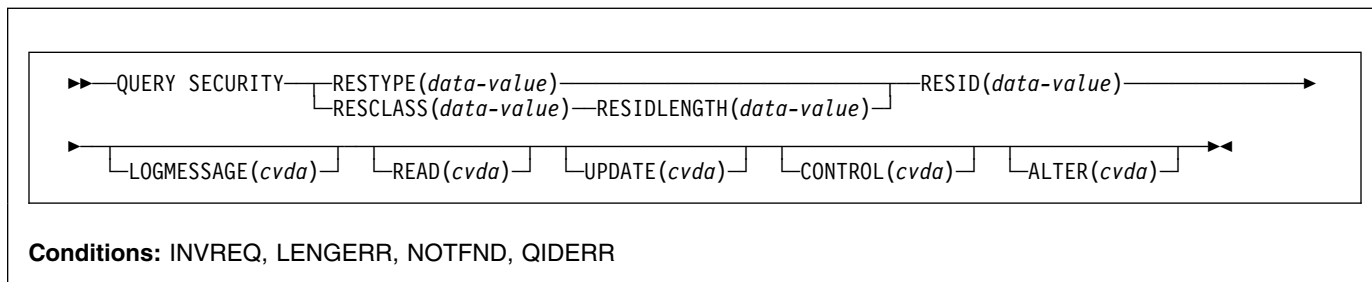
Note: When a CICS program uses EXEC CICS LINK to call another CICS program, the HANDLE CONDITION options are NOT inherited by the linked-to program, but CICS will search preceding logical levels for a HANDLE ABEND exit. See *CICS Application Programming Guide* for further details about the relationship between LINK and HANDLE ABEND.

You can nest PUSH HANDLE ... POP HANDLE command sequences within a task. Each PUSH HANDLE command stacks a set of specifications.

The C language does not support PUSH HANDLE.

QUERY SECURITY

To query the security authorization of the user.



Description

QUERY SECURITY allows the application to determine whether the user has access to resources defined in the external security manager (ESM). These resources can be:

- In CICS resource classes
- In user-defined resource classes

The user in this context is the user invoking the transaction that contains the QUERY SECURITY command.

For more information on the use of the QUERY SECURITY command, see the *CICS Security Guide*.

Options

ALTER(cvda)

enables you to query whether the user has ALTER authority for the named resource. The cvda values returned by CICS are ALTERABLE and NOTALTERABLE.

CONTROL(cvda)

enables you to query whether the user has CONTROL authority for the named resource. The cvda values returned by CICS are CTRLABLE and NOTCTRLABLE.

LOGMESSAGE(cvda)

enables you to inhibit security violation messages. The values passed to CICS are LOG (the default value), or, to inhibit messages, NOLOG.

READ(cvda)

enables you to query whether the user has READ authority command for the named resource. The cvda values returned by CICS are READABLE and NOTREADABLE. READ access authority usually permits nondestructive use of a resource as, for example, in the case of READ and INQUIRE commands.

RESCLASS(data-value)

specifies an 8-character field identifying the name of a valid resource class, that could be non-CICS, in the

ESM. The class name identified by RESCLASS is treated literally with no translation. RESCLASS enables you to define more narrowly the authorization to be queried; for example, you can query at the record or field level.

The responses returned by the command reflect the definition of the RESID resource as defined in the specified RESCLASS.

RESID(data-value)

specifies the name of the particular CICS or user-defined resource that you want to query the user's access to. The value is a character string of 1–12 characters for a CICS resource, and 1–246 characters for a user-defined resource, unless you are using the COBOL3 translator option, when the maximum length is 160 characters.

Note: RESID refers to a CICS-defined resource only when RESTYPE('SPCOMMAND') is specified, otherwise it refers to a user-defined resource. For a list of the CICS RESID values that you can use when RESTYPE('SPCOMMAND') is specified, see the *CICS Security Guide*.

Note that the actual resource checked depends on whether RESCLASS or RESTYPE is specified in the command and whether prefixing is active (SECPRFX=YES specified as a system initialization parameter).

If RESCLASS is specified, the resource checked is always the actual RESID data-value, whether or not prefixing is on or off. IF RESTYPE is specified and SECPRFX=NO, the resource checked is the RESID data-value as specified, however if SECPRFX=YES, the resource checked is the RESID data-value prefixed with the CICS region userid.

RESIDLENGTH(data-value)

specifies the length, as a fullword binary, of the resource identifier in RESID. You only use this parameter when specifying the RESCLASS option.

RESTYPE(*data-value*)

specifies the type of resource (1–12 characters) you want to query the user’s access to.

The responses returned by the command reflect the results that would be obtained if an actual attempt was made to access the specified CICS resource. The value you specify for RESTYPE must be one of the following resource types:

FILE	
JOURNALNUM	
PROGRAM	
PSB	
SPCOMMAND ¹	
TDQUEUE	
TRANSACTION	
TRANSATTACH	
TSQUEUE	
¹ See <i>CICS Security Guide</i> .	

With dynamic transaction routing, it is not necessary to install transaction definitions in terminal owning regions. A QUERY SECURITY command with a RESTYPE of TRANSATTACH returns the NOTFND condition if the transaction is not installed. Programmers, however, should be aware that the transaction may be routed dynamically.

UPDATE(*cvda*)

enables you to query whether the user has UPDATE authority for the named resource. The cvda values returned by CICS are UPDATABLE and NOTUPDATABLE. UPDATE access authority usually permits destructive use of a resource as, for example, in the case of WRITE, DELETE, or UPDATE commands.

Conditions

INVREQ

RESP2 values:

- 7 The cvda value is not valid for the LOGMESSAGE.
- 9 The RESID is invalid or filled with blanks.
- 10 The external security manager (ESM) is inactive or not present.

Default action: terminate the task abnormally.

LENGERR

RESP2 values:

- 6 The RESIDLENGTH value is not valid, that is, not in the range 1 through 246.

Default action: terminate the task abnormally.

NOTFND

RESP2 values:

- 1 The RESID is not valid.
- 2 The RESTYPE is not valid.
- 3 The RESID value for RESTYPE (SPCOMMAND) not valid.
- 5 The RESCLASS is not defined to the external security manager (ESM).
- 8 The resource is not protected. This is only returned when QUERY SECURITY is used with the RESCLASS option (and never occurs with RESTYPE).

Possible causes include:

- RESCLASS not active.
- No profile found.
- ESM not active.

Default action: terminate the task abnormally.

QIDERR

RESP2 values:

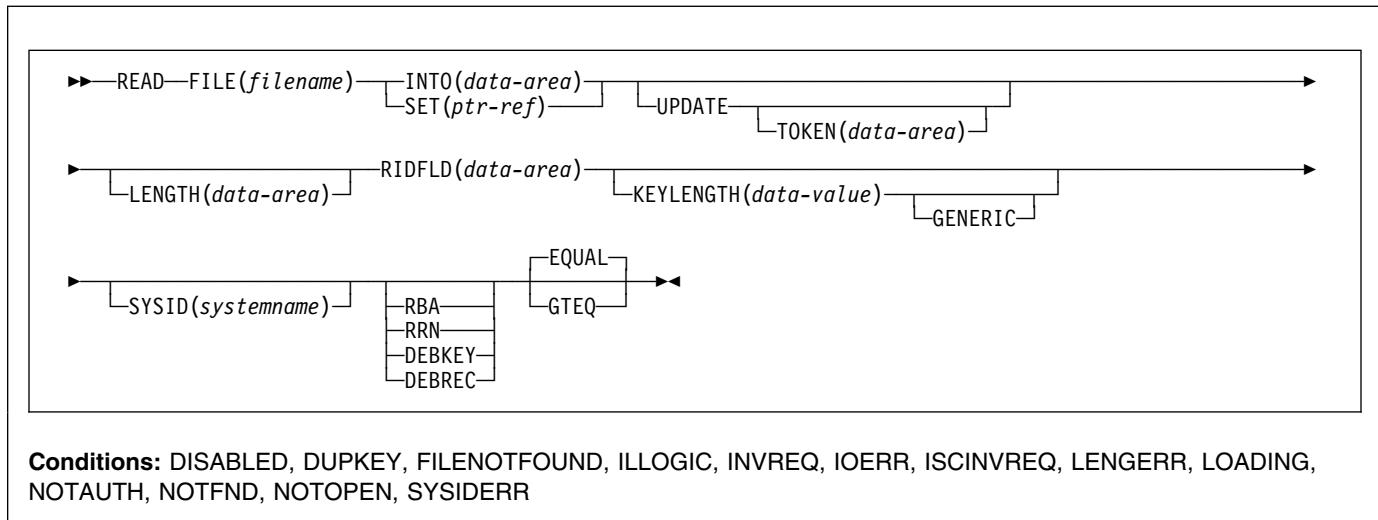
- 1 An indirect queue name associated with the given RESID is not found.

Default action: terminate the task abnormally.

READ

READ

Read a record from a file.



Description

READ reads a record from a file on a local or a remote system.

For both UPDATE and non-UPDATE commands, you must identify the record to be retrieved by the record identification field specified in the RIDFLD option. Immediately upon completion of a READ UPDATE command, the RIDFLD data area is available for reuse by the application program.

When this command reads a CICS-maintained data table, a READ request with UPDATE or RBA is always satisfied by a call to VSAM. A full key read that is neither a generic read nor a READ UPDATE, is satisfied by a reference to the data table if possible. If the record is not found in the table, the source data set is accessed, unless the table is known to be complete, that is, all records in the source are also present in the table (which is the case if loading is finished and none has been rejected by a user exit).

When this command reads a user-maintained data table, only the data table is accessed once loading is complete; the VSAM file is not referenced in any way.

Options

DEBKEY

(blocked DAM) specifies that deblocking is to occur by key. If neither DEBREC nor DEBKEY is specified, deblocking does not occur.

DEBREC

(blocked DAM) specifies that deblocking is to occur by relative record (relative to zero). If neither DEBREC nor DEBKEY is specified, deblocking does not occur.

EQUAL

specifies that the search is satisfied only by a record having the same key (complete or generic) as that specified in the RIDFLD option.

FILE(*filename*)

specifies the of the file to be accessed.

If SYSID is specified, the data set to which this file refers is assumed to be on a remote system irrespective of whether the name is defined in the FCT. Otherwise, the FCT entry is used to find out whether the data set is on a local or a remote system.

GENERIC

(VSAM KSDS, paths and data tables) specifies that the search key is a generic key whose length is specified in the KEYLENGTH option. The search for a record is satisfied when a record is found that has the same starting characters (generic key) as those specified.

GTEQ

(VSAM KSDS, paths and data tables) specifies that, if the search for a record having the same key (complete or generic) as that specified in the RIDFLD option is unsuccessful, the first record having a greater key is retrieved.

INTO(*data-area*)

specifies the data area into which the record retrieved from the data set is to be written.

When INTO is specified, LENGTH must either be specified explicitly or must be capable of being defaulted from the INTO option using the length attribute reference in assembler language, or STG and CSTG in PL/I. LENGTH must be specified explicitly in C.

KEYLENGTH(*data-value*)

specifies the length (halfword binary) of the key that has been specified in the RIDFLD option, except when RBA or RRN is specified, in which case KEYLENGTH is not valid. This option must be specified if GENERIC is specified, and it can be specified whenever a key is specified. However, if the length specified is different from the length defined for the data set and the operation is not generic, the INVREQ condition occurs.

The INVREQ condition also occurs if GENERIC is specified and the KEYLENGTH is not less than that specified in the VSAM definition.

If KEYLENGTH(0) is used with the object of reading the first record in the data set, the GTEQ option must also be specified. If EQUAL is specified either explicitly or by default with KEYLENGTH(0), the results of the READ are unpredictable.

LENGTH(*data-area*)

specifies the length, as a halfword binary value, of the data area where the record is to be put. On completion of the READ command, the LENGTH parameter contains the actual length of the record.

LENGTH must be specified with the INTO option on READ commands involving variable-length records. It need not be specified for fixed-length records, but its inclusion is recommended because:

- It causes a check to be made that the record being read is not too long for the available data area
- When reading fixed-length records into an area longer than the record being accessed, the LENGERR condition is raised if the LENGTH option is not specified. If the length specified exceeds the file record length, CICS uses the longer length for the move. If the target area in the application program is not large enough, storage is overlaid beyond the target area.

See “LENGTH options” on page 5 for more information about when LENGTH must be specified.

When reading into a target data area longer than the record being read, the contents of the target data area, from the end of the retrieved record to the end of the target data area, are unpredictable.

If you specify the INTO option, the LENGTH argument must be a data area that specifies the largest record the program accepts. If the retrieved record is longer than the value specified in the LENGTH option, the record is truncated to the specified value and the LENGERR condition occurs. In this case, the LENGTH data area is set to the length of the record prior to truncation.

If you specify the SET option, the LENGTH option need not be specified but, if it is, the argument must be a data area.

RBA

(VSAM KSDS or ESDS base data sets, or CICS-maintained data tables only, but not paths) specifies that the record identification field specified in the RIDFLD option contains a relative byte address. This option should be used only when reading records from an ESDS base or when reading from a KSDS base and using relative byte addresses instead of keys to identify the records.

You cannot use RBA for:

- User-maintained data tables
- KSDS files that hold more than 4GB

RIDFLD(*data-area*)

specifies the record identification field. The contents can be a key, a relative byte address, or relative record number (for VSAM data sets), or a block reference, a physical key, and a deblocking argument (for DAM data sets). For a relative byte address or a relative record number, the format of this field must be fullword binary. For a relative byte address, the RIDFLD can be greater than or equal to zero. For a relative record number, the RIDFLD can be greater than or equal to 1.

Immediately upon completion of the command, the RIDFLD data area is available for reuse by the application program, even if UPDATE was specified.

Make sure that the variable specified by RIDFLD is not shorter than the KEYLENGTH specified in this command or, if KEYLENGTH is not specified, the key length of the file you are reading; otherwise, the results are unpredictable.

RRN

(VSAM RRDS) specifies that the record identification field specified in the RIDFLD option contains a relative record number. This option should only be used with files referencing relative record data sets.

SET(*ptr-ref*)

indicates that CICS is to supply a buffer where the record is read, and specifies the pointer reference that is to contain the address of the retrieved record.

If the DUPKEY condition occurs in assembler language the specified register has not been set. The specified register can be loaded from DFHEITP1.

The pointer reference is valid until the next READ command for the same file or until completion of a corresponding REWRITE, DELETE, or UNLOCK command, or a SYNCPOINT in the case of READ UPDATE SET. If you want to retain the data within the field addressed by the pointer, it should be moved to your own area.

If DATALOCATION(ANY) is specified in the RDO PROGRAM resource definition for the application

READ

program, the address of the data can be above or below the 16MB line.

If DATALOCATION(BELOW) is specified in the RDO PROGRAM resource definition for the application program, the address of the data is below the 16MB line.

If TASKDATAKEY (USER) is specified in the RDO TRANSACTION resource definition for the executing transaction, and storage protection is active, the data returned is in user-key DSA (UDSA or EUDSA); otherwise the data is returned in a CICS-key DSA (CDSA or ECDSA).

SYSID(*systemname*)

specifies the name of the system the request is directed to.

If you specify SYSID, and omit both RBA and RRN, you must also specify LENGTH and KEYLENGTH; they cannot be found in the FCT.

TOKEN(*data-area*)

specifies, as a fullword binary value, a unique identifier for this READ UPDATE request. This is an output value returned by file control to the requesting task, for use in associating a subsequent REWRITE or DELETE (or UNLOCK) request with the record returned on this READ UPDATE request.

Note: If you specify TOKEN it implies update.

UPDATE

specifies that the record is to be obtained for updating or (for VSAM and data tables) deletion. If this option is omitted, a read-only operation is assumed.

UPDATE guarantees read integrity.

Conditions

Note: RESP2 values are not set for files that are on remote systems.

DISABLED

RESP2 values:

50 A file is disabled because it was initially defined as disabled and has not since been enabled.

A file is disabled because it has been disabled by a SET FILE or a CEMT SET FILE command.

Default action: terminate the task abnormally.

DUPKEY

RESP2 values: (VSAM)

140 A record is retrieved by way of an alternate index with the NONUNIQUEKEY attribute, and another alternate index record with the same key follows

In assembler language, if the SET option is being used, the specified register has not been set, but can be loaded from DFHEITP1.

Default action: terminate the task abnormally.

FILENOTFOUND

RESP2 values:

1 The file name supplied in the FILE option cannot be found in the FCT.

Default action: terminate the task abnormally.

ILLOGIC

RESP2 values: (VSAM)

110 A VSAM error occurs that does not fall within one of the other CICS response categories.

(See EIBRCODE in the description of the EXEC interface block, starting on 373.)

For user-maintained data tables, this condition occurs only for a non-UPDATE READ during loading when CICS has attempted to retrieve the record from the source data set.

Default action: terminate the task abnormally.

INVREQ

RESP2 values:

20 READ is not allowed according to the file entry specification in the FCT.

A READ command with the UPDATE option is issued to a file where update operations are not allowed according to the file entry specification in the FCT.

25 The KEYLENGTH and GENERIC options are specified, and the length specified in the KEYLENGTH option is greater than or equal to the length of a full key.

26 The KEYLENGTH option is specified (but the GENERIC option is not specified), and the specified length does not equal the length defined for the data set to which this file refers.

40 A DAM key conversion error occurred.

42 The KEYLENGTH and GENERIC options are specified, and the length specified in the KEYLENGTH option is less than zero.

44 The command does not conform to the format of READ for a user-maintained data table; for example if RBA is specified.

48 An attempt is made to function ship an UPDATE request that includes a TOKEN.

49 An attempt was made to read a file that is an invalid file type.

Default action: terminate the task abnormally.

IOERR

RESP2 values:

120 There is an I/O error during the READ operation. An I/O error is any unusual event that is not covered by a CICS condition.

For VSAM files, IOERR normally indicates a hardware error.

For user-maintained data tables, this condition occurs only for a non-UPDATE READ during loading when CICS has attempted to retrieve the record from the source data set.

(Further information is available in the EXEC interface block, defined on 373.)

Default action: terminate the task abnormally.

ISCVREQ

RESP2 values:

- 70** The remote system indicates a failure that does not correspond to a known condition.

Default action: terminate the task abnormally.

LENGERR

RESP2 values:

- 10** Neither the LENGTH option nor the SET option is specified on a READ command for a file with variable-length records or for a DAM file with variable-length or undefined-format records.
- 11** The length of a record read with the INTO option specified exceeds the value specified in the LENGTH option; the record is truncated, and the data area supplied in the LENGTH option is set to the actual length of the record.
- 13** An incorrect length is specified for a file with fixed-length records.

Default action: terminate the task abnormally.

LOADING

RESP2 values:

- 104** A READ UPDATE is issued for a user-maintained data table that is currently being loaded. A user-maintained data table cannot be modified during loading.

A READ GENERIC or a READ GTEQ is issued for a user-maintained data table that is currently being loaded. While a user-maintained data table is being loaded, you can use only read requests with precise keys.

Default action: terminate the task abnormally.

NOTAUTH

RESP2 values:

- 101** A resource security check has failed on FILE(filename).

Default action: terminate the task abnormally.

NOTFND

RESP2 values:

- 80** An attempt to retrieve a record based on the search argument provided is unsuccessful.

For user-maintained data tables, this condition occurs if an attempt to read a record is unsuccessful because there is no entry with the specified key in the data table. This does not mean that there is no such record in the source data set; it may be that such a record is present but was either rejected during initial loading by the user exit XDTRD, or was subsequently deleted from the user-maintained data table.

Default action: terminate the task abnormally.

NOTOPEN

RESP2 values:

- 60** The requested file is CLOSED and UNENABLED. The CLOSED, UNENABLED state is reached after a CLOSE request has been received against an OPEN ENABLED file and the file is no longer in use. This state can also be specified as the initial state by means of the FILSTAT parameter of the DFHFCT TYPE=FILE macro, or the STATUS(UNENABLED) and OPENTIME(FIRSTREF) options on the RDO FILE resource definition.

The requested file is OPEN and in use by other transactions, but a CLOSE request against the file has been received.

This condition does not occur if the request is made to either a CLOSED, ENABLED file or to a CLOSED, DISABLED file. In the first case, the file is opened as part of executing the request. In the second case, the DISABLED condition occurs.

Default action: terminate the task abnormally.

SYSIDERR

RESP2 values:

- 130** The SYSID option specifies a name that is neither the local system nor a remote system (made known to CICS by defining a CONNECTION). SYSIDERR also occurs when the link to the remote system is closed.

Default action: terminate the task abnormally.

Examples

The following example shows you how to read a record from a file named MASTER into a specified data area:

```
EXEC CICS READ
      INTO(RECORD)
      FILE('MASTER')
      RIDFLD(ACCTNO)
```

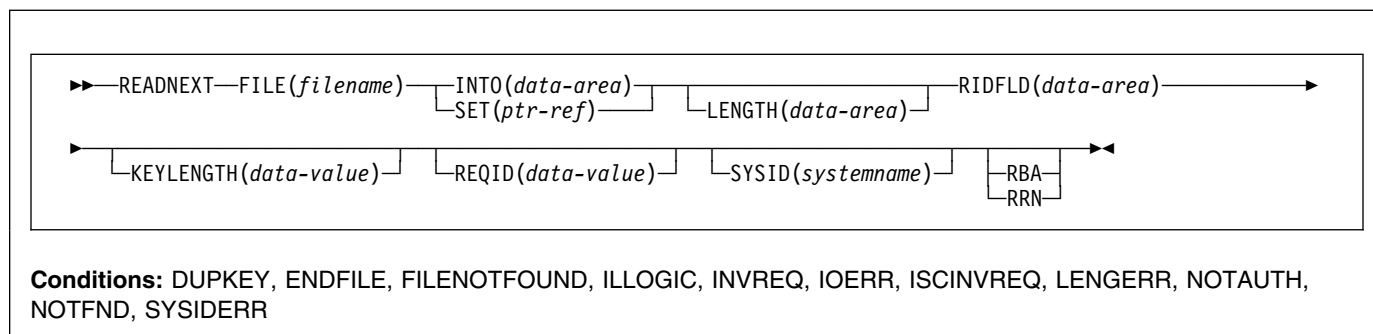
READ

The following example shows you how to read a record for update from a VSAM file using a generic key and specifying a greater-or-equal key search.

```
EXEC CICS READ  
  INTO(RECORD)  
  LENGTH(RECLEN)  
  FILE('MSTVSAM')  
  RIDFLD(ACCTNO)  
  KEYLENGTH(4)  
  GENERIC  
  GTEQ  
  UPDATE
```

READNEXT

Read next record during a browse of a file.



Description

READNEXT can be used repeatedly to read records in sequential order from a file on a local or a remote system. Such a series of sequential read commands is known as a **browse** of the file. A browse can also consist of a sequence of READNEXT and READPREV commands in any order. A browse must be initiated with the STARTBR command, to identify the starting point of the browse, and terminated with the ENDBR command.

You must provide, in the RIDFLD option, a data area that is sufficiently large to contain a complete identifier (full key, RBA, or RRN) of records in the file. This data area can be used both as an output and as an input parameter.

It is used as an output parameter when CICS, on completion of each READNEXT command, places the complete identifier of the record just retrieved into the RIDFLD data area. CICS then holds this identifier to mark the point from which the subsequent READNEXT is to continue.

It may, except for DAM, also be used as an input parameter. Modifying the RIDFLD before issuing the next READNEXT command causes that command to reposition the browse to the new identifier, from which it continues in the usual way. If the browse was started with the GENERIC option, the modified RIDFLD must be generic. If the browse was started with the GTEQ option, the next record returned is the first record in the data set with a key greater than or equal to the modified RIDFLD.

A READNEXT command following a READPREV, or a STARTBR or RESETBR that specified a 'last' key value, is treated as though the RIDFLD value has been modified, and results in a reposition (as above).

Options

FILE(filename)

specifies the name of the file to be browsed.

If SYSID is specified, the data set to which this file refers is assumed to be on a remote system irrespective of whether the name is defined in the FCT. Otherwise, the FCT entry is used to find out whether the data set is on a local or a remote system.

INTO(data-area)

specifies the data area into which the record retrieved from the data set is to be written.

KEYLENGTH(data-value)

specifies the length (halfword binary) of the key that has been specified in the RIDFLD option, except when RBA or RRN is specified, in which case KEYLENGTH is not valid.

If the browse was started without the GENERIC option (that is a full key browse) and if the length specified is different from the length defined for the data set, the INVREQ condition occurs.

If the browse was started with the GENERIC option (that is, a generic key browse), and if the length specified is greater than the length specified for the data set, the INVREQ condition occurs.

If GTEQ and GENERIC were specified on the most recent STARTBR or RESETBR command, issuing READNEXT with KEYLENGTH(0) specified repositions the BROWSE at the start of the file. If EQUAL had been specified, the effect of READNEXT KEYLENGTH(0) would be unpredictable.

For a generic browse, CICS maintains a current key length for the browse. The current key length is initialized to be the value specified as KEYLENGTH on the STARTBR command.

The current key length may be modified by specifying KEYLENGTH on a READNEXT or RESETBR command.

READNEXT

If the current key length is changed, this causes the browse to be repositioned. The browse is repositioned to the key whose initial characters match the value specified in the RIDFLD for the current key length.

The current key length is zero after a request that specifies KEYLENGTH(0).

IF KEYLENGTH is omitted on a READNEXT command, the current key length remains the same and the browse continues without repositioning.

If KEYLENGTH is specified on a READNEXT command and is equal to the current key length, this is treated as being no change and the browse is not repositioned. The one exception to this is when KEYLENGTH(0) is specified, which always causes the browse to be repositioned to the beginning of the file.

KEYLENGTH can be specified during a generic browse with a value equal to the full key length. This does not cause the current key length to change and the browse is not repositioned. The ability to specify the full key length during a generic browse is provided to allow requests that specify SYSID to be able to tell the function-shipping transformers how long the key is, so that the transformers can ship the key to the file-owning region.

A browse can be repositioned by modifying the RIDFLD data area. A generic browse is repositioned only if the modification to RIDFLD changes the part of RIDFLD corresponding to the current key length. A consequence of this is that the browse can not be repositioned by modifying the RIDFLD data area if the current key length is zero.

LENGTH(*data-area*)

specifies the length, as a halfword binary value, of the record to be retrieved. On completion of a retrieval operation, the LENGTH parameter contains the actual length of the retrieved record.

This option must be specified if SYSID is specified. It must also be specified with the INTO option on commands involving variable-length records. It need not be specified for fixed-length records, but its inclusion is recommended because:

- It causes a check to be made that the record being read is not too long for the available data area.
- When browsing fixed-length records into an area longer than the record being accessed, the LENGERR condition is raised if the LENGTH option is not specified. If the length specified exceeds the file record length, CICS uses the longer length for the move. If the target area in the application program is not large enough, storage is overlaid beyond the target area.

See “LENGTH options” on page 5 for more information about when LENGTH must be specified.

When browsing into a target data area longer than the record being read, the contents of the target data area,

from the end of the retrieved record to the end of the target data area, are unpredictable.

If you specify the INTO option, the LENGTH argument must be a data area that specifies the largest record the program accepts. If the retrieved record is longer than the value specified in the LENGTH option, the record is truncated to the specified value and the LENGERR condition occurs. In this case, the LENGTH data area is set to the length of the record prior to truncation.

If you specify the SET option, the LENGTH option need not be specified but, if it is, the argument must be a data area.

Note that a file control command issued against a variable-length record in a file defined on the local CICS system fails with a LENGERR condition if a length is not specified. However, if the same command is issued against a file defined on a remote system, the command does not fail. The transformer which passes CICS commands from one CICS region to another has to pass a default value for the length argument if a value is not specified. This is because, if the file is remote, the transformer cannot identify the record as fixed or variable, and it is valid not to pass a length when issuing a request against a fixed-length record.

RBA

(VSAM KSDS or ESDS base data sets, or CICS-maintained data tables only, but not paths) specifies that the record identification field specified in the RIDFLD option contains a relative byte address.

This option must be specified when the STARTBR or RESETBR command specified the RBA option.

You cannot use RBA for:

- User-maintained data tables
- KSDS files that hold more than 4GB.

REQID(*data-value*)

specifies as a halfword binary value a unique request identifier for a browse, used to control multiple browse operations on a file. If this option is not specified, a default value of zero is assumed.

RIDFLD(*data-area*)

specifies the record identification field. The contents can be a key, a relative byte address, or relative record number (for VSAM data sets), or a block reference, physical key, and deblocking argument (for DAM data sets). For a relative byte address or a relative record number, the format of this field must be fullword binary. For a relative byte address, the RIDFLD can be greater than or equal to zero. For a relative record number, the RIDFLD can be greater than or equal to 1.

Even if the browse is generic, this field should always be large enough to accommodate the complete record identifier. This is because, on completion of the READNEXT command, the field is updated by CICS with the complete identification of the record retrieved.

RRN

(VSAM RRDS) specifies that the record identification field specified in the RIDFLD option contains a relative record number.

SET(ptr-ref)

specifies the pointer reference that is to be set to the address of the retrieved record.

In assembler language, if the DUPKEY condition occurs, the register specified will not have been set, but can be loaded from DFHEITP1.

The pointer reference is valid until the next READNEXT or READPREV command specifying SET for the same browse (REQID) for the same file. The pointer is no longer valid after an ENDBR or SYNCPOINT command. If you want to retain the data within the field addressed by the pointer, you must move it to your own area.

If DATALOCATION(ANY) is specified in the RDO PROGRAM resource definition for the application program, the address of the data can be above or below the 16MB line.

If DATALOCATION(BELOW) is specified in the RDO PROGRAM resource definition for the application program, the address returned in the SET pointer is below the 16MB line.

If TASKDATAKEY(USER) is specified in the RDO TRANSACTION resource definition for the executing transaction, and storage protection is active, the data returned is in user-key DSA (UDSA or EUDSA); otherwise the data is returned in a CICS-key DSA (CDSA or ECDSA).

SYSID(systemname)

specifies the name of the system to which the request is directed.

If you specify SYSID, and omit both RBA and RRN, you must also specify LENGTH and KEYLENGTH; they cannot be found in the FCT.

Conditions

Note: RESP2 values are not set for files that are on remote systems.

DUPKEY

RESP2 values (VSAM):

140 A record is retrieved by way of an alternate index with the NONUNIQUEKEY attribute, and another alternate index record with the same key follows. It does not occur as a result of a READNEXT command that reads the last of the records having the nonunique key.

In assembler language, if the SET option is being used, the register specified will not have been set, but can be loaded from DFHEITP1.

Default action: terminate the task abnormally.

ENDFILE

RESP2 values:

90 An end-of-file condition is detected during the browse.

Default action: terminate the task abnormally.

FILENOTFOUND

RESP2 values:

1 A file name referred to in the FILE option cannot be found in the FCT.

Default action: terminate the task abnormally.

ILLOGIC

RESP2 values (VSAM):

110 A VSAM error occurs that does not fall within one of the other CICS response categories.
(See EIBRCODE in the EXEC interface block; starting on 373.)

Default action: terminate the task abnormally.

INVREQ

RESP2 values:

25 The KEYLENGTH option is specified for a generic browse (that is one where GENERIC was specified on the STARTBR or the last RESETBR) and the value of KEYLENGTH was greater than the full key length.

26 The KEYLENGTH option is specified for a nongeneric browse, and the specified length does not equal the length defined for the data set to which this file refers.

34 The REQID, if any, does not match that of any successful STARTBR command.

37 The type of record identification (for example, key or relative byte address) used to access a data set during the browse is changed by the READNEXT command.

42 The KEYLENGTH option is specified for a generic browse (that is one where GENERIC was specified on the STARTBR or the last RESETBR) and the value of KEYLENGTH was less than zero.

49 An attempt was made to browse a file that is an invalid file type.
#

Default action: terminate the task abnormally.

IOERR

RESP2 values:

120 There is an I/O error during the READNEXT command. An I/O error is any unusual event that is not covered by a CICS condition.

For VSAM files, IOERR normally indicates a hardware error.

READNEXT

(Further information is available in the EXEC interface block, defined on 373.)

Default action: terminate the task abnormally.

ISCINVREQ

RESP2 values:

- 70** The remote system indicates a failure that does not correspond to a known condition.

Default action: terminate the task abnormally.

LENGERR

RESP2 values:

- 10** Neither the LENGTH nor the SET option is specified for a file with variable-length records or a DAM file with undefined-format records.
- 11** The length of the record read with the INTO option specified exceeds the value specified in the LENGTH option; the record is truncated, and the data area supplied in the LENGTH option is set to the actual length of the record.
- 13** An incorrect length is specified for a file with fixed-length records.

Default action: terminate the task abnormally.

NOTAUTH

RESP2 values:

- 101** A resource security check has failed on FILE(filename).

Default action: terminate the task abnormally.

NOTFND

RESP2 values:

- 80** An attempt to retrieve a record based on the search argument provided is unsuccessful. This may occur if the READNEXT command immediately follows a STARTBR command that specified the key of the last record in the data set (a complete key of X'FF's).

Default action: terminate the task abnormally.

SYSIDERR

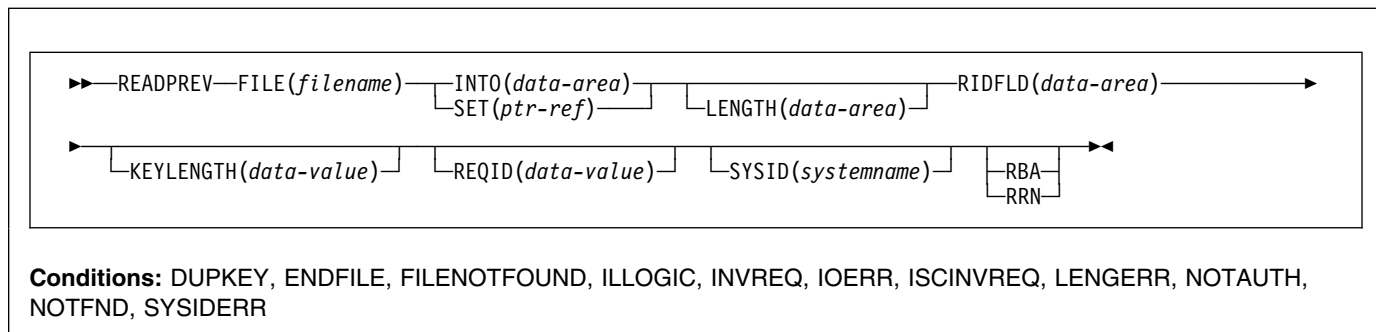
RESP2 values:

- 130** The SYSID option specifies a name that is neither the local system nor a remote system (made known to CICS by defining a CONNECTION). SYSIDERR also occurs when the link to the remote system is closed

Default action: terminate the task abnormally.

READPREV

Read previous record during a file browse; VSAM and data tables only.



Description

READPREV can be used repeatedly to read records in reverse sequential order from a VSAM file on a local or a remote system.

Such a series of sequential read commands is known as a **browse** of the file. A browse may also consist of a sequence of READNEXT and READPREV commands in any order. A browse must be initiated with the STARTBR command, to identify the start of the browse, and terminated with the ENDBR command.

You must provide, in the RIDFLD option, a data area that is sufficiently large to contain a complete identifier (full key, RBA, or RRN) of records in the file. This data area is used both as an output and as an input parameter.

It is used as an output parameter when CICS, on completion of each READPREV command, places the complete identifier of the record just retrieved into the RIDFLD data area. CICS then holds this identifier to mark the point from which the subsequent READPREV is to continue.

It may also be used as an input parameter. Modifying the RIDFLD before issuing the next READPREV command causes that command to reposition the browse to the new identifier, from which it continues in the usual way. The modified record identifier must always be a full key, RBA, or RRN. A generic key may not be specified, nor may a browse that was started with the GENERIC option include a READPREV command.

If you include a READPREV command immediately following a STARTBR command, your STARTBR command RIDFLD must specify the key of a record that exists on the data set; otherwise the NOTFND condition will occur.

A READPREV command following a READNEXT, or a STARTBR or RESETBR that did not specify a 'last' key

value, is treated as though the RIDFLD value had been modified and results in a reposition (as above).

Options

FILE(filename)

specifies the of the file being browsed.

If SYSID is specified, the data set to which this file refers is assumed to be on a remote system irrespective of whether the name is defined in the file resource definition. Otherwise, the file definition is used to find out whether the data set is on a local or a remote system.

INTO(data-area)

specifies the data area into which the record retrieved from the data set is to be written.

KEYLENGTH(data-value)

specifies the length (halfword binary) of the key that has been specified in the RIDFLD option, except when RBA or RRN is specified, in which case KEYLENGTH is not valid. If the length specified is different from the length defined for the data set, the INVREQ condition occurs.

LENGTH(data-area)

specifies the length, as a halfword binary value, of the record to be retrieved. On completion of a retrieval operation, LENGTH contains the actual length of the retrieved record.

This option must be specified if SYSID is specified. It must also be specified with the INTO option on READPREV commands involving variable-length records. It need not be specified for fixed-length records, but its inclusion is recommended because:

- It causes a check to be made that the record being read is not too long for the available data area.
- When browsing fixed-length records into an area longer than the record being accessed, the

READPREV

LENGERR condition is raised if the LENGTH option is not specified. If the length specified exceeds the file record length, CICS uses the longer length for the move. If the target area in the application program is not large enough, storage is overlaid beyond the target area.

See “LENGTH options” on page 5 for more information about when LENGTH must be specified.

When browsing into a target data area longer than the record being read, the contents of the target data area, from the end of the retrieved record to the end of the target data area, are unpredictable.

If you specify the INTO option, LENGTH specifies the largest record the program accepts. If the retrieved record is longer than the value specified in the LENGTH option, the record is truncated to the specified value and the LENGERR condition occurs. In this case, the LENGTH data area is set to the length of the record prior to truncation.

If you specify the SET option, LENGTH need not be specified.

RBA

(VSAM KSDS or ESDS base data sets, or CICS-maintained data tables only, but not paths) specifies that the record identification field specified in the RIDFLD option contains a relative byte address. You must specify this option after any STARTBR or RESETBR command that specified the RBA option.

You cannot use RBA for:

- User-maintained data tables
- KSDS files that hold more than 4GB

REQID(*data-value*)

specifies as a halfword binary value a unique request identifier for a browse, used to control multiple browse operations on a file. If this option is not specified, a default value of zero is assumed.

RIDFLD(*data-area*)

specifies the record identification field. The contents can be a key, a relative byte address, or relative record number. For a relative byte address or a relative record number, the format of this field must be fullword binary. For a relative byte address, the RIDFLD can be greater than or equal to zero. For a relative record number, the RIDFLD can be greater than or equal to 1.

On completion of the READPREV command, this field is updated by CICS with the complete identification of the record retrieved.

RRN

(VSAM RRDS) specifies that the record identification field specified in the RIDFLD option contains a relative record number.

SET(*ptr-ref*)

specifies the pointer reference that is to be set to the address of the retrieved record.

In assembler language, if the DUPKEY condition occurs, the register specified will not have been set, but can be loaded from DFHEITP1.

The pointer reference is valid until the next READPREV or READNEXT command specifying SET for the same browse (REQID) for the same file. The pointer is no longer valid after an ENDBR or SYNCPOINT command. If you want to retain the data within the field addressed by the pointer, you must move it to your own area.

If DATALOCATION(ANY) is specified in the RDO PROGRAM resource definition for the application program, the address returned in the SET pointer can be above or below the 16MB line.

If DATALOCATION(BELOW) is specified in the RDO PROGRAM resource definition for the application program, the address returned in the SET pointer is below the 16MB line.

If TASKDATAKEY(USER) is specified in the RDO TRANSACTION resource definition for the executing transaction, and storage protection is active, the data returned is in user-key DSA (UDSA or EUDSA); otherwise the data is returned in a CICS-key DSA (CDSA or ECDSA).

SYSID(*systemname*)

specifies the name of the system to which the request is directed.

If you specify SYSID, and omit both RBA and RRN, you must also specify LENGTH and KEYLENGTH.

Conditions

Note: RESP2 values are not set for files that are on remote systems.

DUPKEY

RESP2 values:

140 A record is retrieved by way of an alternate index with the NONUNIQUEKEY attribute, and another alternate index record with the same key exists.

In assembler language, if the SET option is being used, the specified register has not been set, but can be loaded from DFHEITP1.

Default action: terminate the task abnormally.

ENDFILE

RESP2 values:

90 An end-of-file condition is detected during a browse.

Default action: terminate the task abnormally.

FILENOTFOUND

RESP2 values:

- 1 A file name referred to in the FILE option cannot be found in the FCT.

Default action: terminate the task abnormally.

ILLOGIC

RESP2 values (VSAM):

- 110 A VSAM error occurs that does not fall within one of the other CICS response categories.

(See EIBRCODE in the EXEC interface block, starting on 373.)

Default action: terminate the task abnormally.

INVREQ

RESP2 values:

- 24 A READPREV command is issued for a file for which the previous STARTBR or RESETBR command has the GENERIC option.
- 26 The KEYLENGTH option is specified and the specified length does not equal the length defined for the data set for this file refers to.
- 37 The type of record identification (for example, key or relative byte address) used to access a data set during the browse is changed.
- 39 A READPREV is issued for a DAM file.
- 41 The REQID, if any, does not match that of any successful STARTBR command.
- # 49 An attempt was made to browse a file that is an invalid file type.

Default action: terminate the task abnormally.

IOERR

RESP2 values:

- 120 An I/O error occurred during the browse. An I/O error is any unusual event that is not covered by a CICS condition.

For VSAM files, IOERR normally indicates a hardware error. (Further information is available in the EXEC interface block, defined on 373.)

Default action: terminate the task abnormally.

ISCVREQ

RESP2 values:

- 70 The remote system indicates a failure that does not correspond to a known condition.

Default action: terminate the task abnormally.

LENGERR

RESP2 values:

- 10 Neither the LENGTH nor the SET option is specified for a file with variable-length records.

- 11 The length of the record read with the INTO option specified exceeds the value specified in the LENGTH option; the record is truncated, and the data area supplied in the LENGTH option is set to the actual length of the record.

- 13 An incorrect length is specified for a file with fixed-length records.

Default action: terminate the task abnormally.

NOTAUTH

RESP2 values:

- 101 A resource security check has failed on FILE(filename).

Default action: terminate the task abnormally.

NOTFND

RESP2 values:

- 80 An attempt to retrieve a record based on the search argument provided is unsuccessful. One place where this may occur is where the READPREV command immediately follows a STARTBR or RESETBR command that specified GTEQ and the key of a record that does not exist on the data set.

Default action: terminate the task abnormally.

SYSIDERR

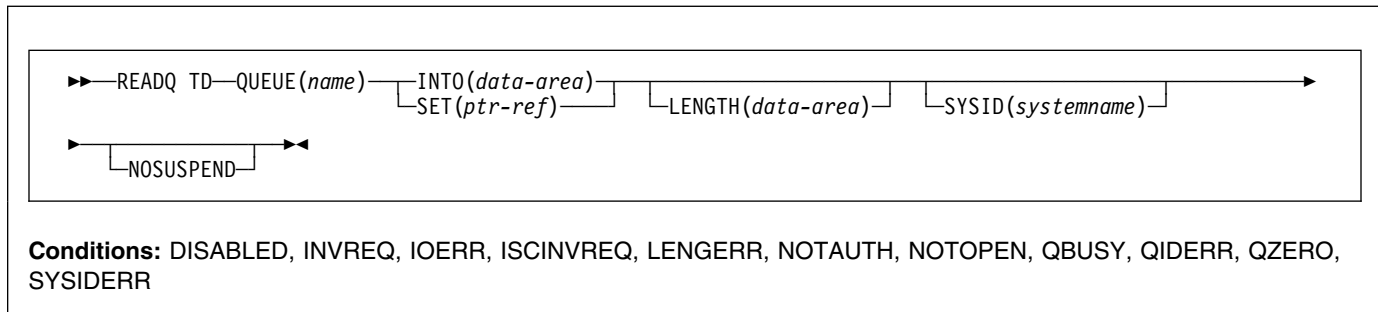
RESP2 values:

- 130 The SYSID option specifies a name that has not been defined to CICS as a remote system (by defining an RDO CONNECTION). SYSIDERR also occurs when the link to the remote system is closed.

Default action: terminate the task abnormally.

READQ TD

Read data from the transient data queue.



Description

READQ TD reads transient data from a queue (after which the record is no longer available).

If you are using automatic transaction initiation (ATI) (see the section on ATI in the *CICS Application Programming Guide* for introductory information) your application should test for the QZERO condition to ensure that termination of an automatically initiated task occurs only when the queue is empty.

If the READQ TD command attempts to access a record in a logically recoverable intrapartition queue that is being written to, or deleted by, another task, and there are no more committed records, the command waits until the queue is no longer being used for output. If, however, the NOSUSPEND option has been specified, or there is an active HANDLE CONDITION for QBUSY, the QBUSY condition is raised.

Options

INTO(*data-area*)

specifies the user data area into which the data read from the transient data queue is to be placed.

LENGTH(*data-area*)

specifies the length, as a halfword binary value, of the record to be read.

If you specify the INTO option, LENGTH specifies the maximum length of data that the program accepts. If the value specified is less than zero, zero is assumed. If the length of the data exceeds the value specified, the data is truncated to that value and the LENGERR condition occurs. On completion of the retrieval operation, the data area is set to the original length of the data record read from the queue.

If you specify the INTO option, LENGTH need not be specified if the length can be generated by the compiler from the INTO variable. See "LENGTH options" on

page 5 for more information about when LENGTH must be specified.

NOSUSPEND

specifies that if the application program attempts to read from a queue that is already being used for output, the task is not suspended until the queue becomes available. Instead, the QBUSY condition is raised.

Note, however, that if a HANDLE CONDITION for QBUSY is active when the command is executed, this also overrides the default action, and control is passed to the user label supplied in the HANDLE CONDITION. This takes precedence over the NOSUSPEND option but is, of course, negated by either NOHANDLE or RESP.

This option only applies to intrapartition queues.

QUEUE(*name*)

specifies the symbolic name (1–4 alphanumeric characters) of the queue to be read from. The name must have been defined in the DCT.

If SYSID is specified, the queue is assumed to be on a remote system irrespective of whether or not the name is defined as remote in the DCT. Otherwise the entry in the DCT is used to find out whether the data set is on a local or a remote system.

SET(*ptr-ref*)

specifies a pointer reference that is to be set to the address of the data read from the queue. CICS acquires an area large enough to hold the record and sets the pointer reference to the address of that area. The area is retained until another transient data command is executed. The pointer reference, unless changed by other commands or statements, is valid until the next READQ TD command or the end of task.

If DATALOCATION(ANY) is specified in the RDO PROGRAM resource definition for the application program, the address of the data can be above or below the 16MB line.

If `DATALOCATION(BELOW)` is specified in the RDO PROGRAM resource definition for the application program, and the data resides above the 16MB line, the data is copied below the 16MB line, and the address of this copy is returned.

If `TASKDATAKEY(USER)` is specified in the RDO TRANSACTION resource definition for the executing transaction, and storage protection is active, the data returned is in user-key DSA (UDSA or EUDSA); otherwise the data is returned in a CICS-key DSA (CDSA or ECDSA)

SYSID(*systemname*)

(remote systems only) specifies the name (1–4 characters) of the system to which the request is directed.

Conditions

DISABLED

occurs when the queue has been disabled.

Default action: terminate the task abnormally.

INVREQ

occurs if READQ names an extrapartition queue that has been opened for output. This condition cannot occur for intrapartition queues.

Default action: terminate the task abnormally.

IOERR

occurs when an input/output error occurs and the data record in error is skipped.

This condition occurs as long as the queue can be read; a QZERO condition occurs when the queue cannot be read.

Default action: terminate the task abnormally.

ISCINVREQ

occurs when the remote system indicates a failure that does not correspond to a known condition.

Default action: terminate the task abnormally.

LENGERR

occurs if READQ names an INTO area that cannot hold all the data that is to be returned to the application. The check is made after the XTDIN exit has been invoked.

Default action: terminate the task abnormally.

NOTAUTH

occurs when a resource security check has failed on `QUEUE(name)`.

Default action: terminate the task abnormally.

NOTOPEN

occurs if the destination is closed. This condition applies to extrapartition queues only.

Default action: terminate the task abnormally.

QBUSY

occurs if a READQ TD command attempts to access a record in a logically recoverable intrapartition queue that is being written to or is being deleted by another task, and there are no more committed records.

The NOSUSPEND option must be specified, or a HANDLE for the condition must be active, for this condition to be raised.

This condition applies only to intrapartition queues.

Default action: ignore the condition.

QIDERR

occurs if the symbolic destination to be used with READQ TD cannot be found.

Default action: terminate the task abnormally.

QZERO

occurs when the destination (queue) is empty or the end of the queue has been reached.

Default action: terminate the task abnormally.

SYSIDERR

occurs when the SYSID option specifies a name that is neither the local system nor a remote system (made known to CICS by defining a CONNECTION).

SYSIDERR also occurs when the link to the remote system is closed.

Default action: terminate the task abnormally.

Examples

The following example shows how to read a record from an intrapartition data set (queue), which in this case is the control system message log (CSML), into a data area specified in the request:

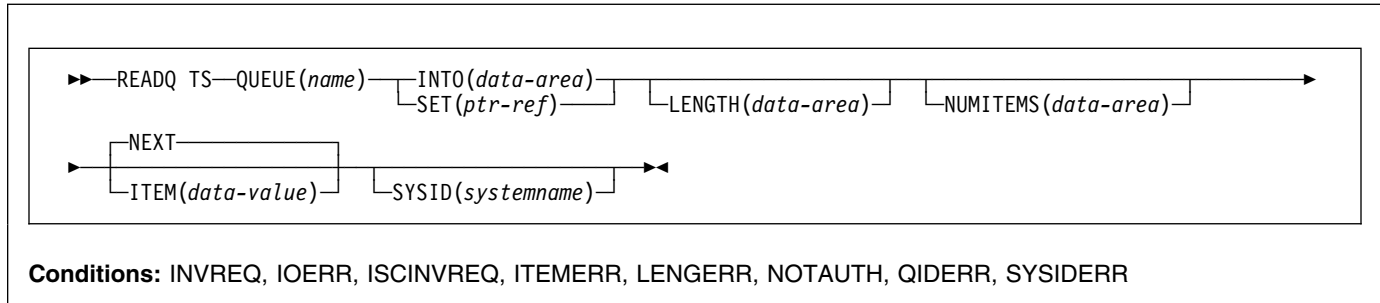
```
EXEC CICS READQ TD
      QUEUE('CSML')
      INTO(DATA)
      LENGTH(LDATA)
```

The following example shows how to read a record from an extrapartition data set (queue) having fixed-length records into a data area provided by CICS; the pointer reference specified by the SET option is set to the address of the storage area reserved for the data record. It is assumed that the record length is known.

```
EXEC CICS READQ TD
      QUEUE(EX1)
      SET(PREF)
```

READQ TS

Read data from temporary storage queue.



Note for dynamic transaction routing

Using this command could create inter-transaction affinities that adversely affect the use of dynamic transaction routing. See the *CICS Application Programming Guide* for more information about transaction affinities.

Description

READQ TS retrieves data from a temporary storage queue in main or auxiliary storage.

Options

INTO(*data-area*)

specifies the data area into which the data is to be written. The data area can be any variable, array, or structure.

ITEM(*data-value*)

provides a halfword binary value that specifies the item number of the logical record to be retrieved from the queue.

LENGTH(*data-area*)

specifies the length, as a halfword binary value, of the record to be read.

If you specify the INTO option, LENGTH need not be specified if the length can be generated by the compiler from the INTO variable.

See “LENGTH options” on page 5 for more information about when LENGTH must be specified.

If you specify INTO, LENGTH defines the maximum length of data that the program accepts. If the value specified is less than zero, zero is assumed. If the length of the data exceeds the value specified, the data is truncated to that value and the LENGERR condition occurs.

On completion of the retrieval operation, the data area is set to the original length of the data record read from the queue.

If you specify the SET option, the LENGTH must be specified.

NEXT

specifies retrieval for the next sequential logical record following the last record retrieved (by any task), or the first record if no previous record has been retrieved.

NUMITEMS(*data-area*)

specifies a halfword binary field into which CICS stores a number indicating how many items there are in the queue. This only occurs if the command completes normally.

QUEUE(*name*)

specifies the symbolic name (1–8 characters) of the queue to be read from.

SET(*ptr-ref*)

specifies the pointer reference that is set to the address of the retrieved data. The pointer reference, unless changed by other commands or statements, is valid until the next READQ TS command or the end of task.

If DATALOCATION(ANY) is specified in the RDO PROGRAM resource definition for the application program, the address of the data can be above or below the 16MB line.

If DATALOCATION(BELOW) is specified in the RDO PROGRAM resource definition for the application program, the address of the data is below the 16MB line.

If TASKDATAKEY(USER) is specified in the RDO TRANSACTION resource definition for the executing transaction, and storage protection is active, the data returned is in user-key DSA (UDSA or EUDSA); otherwise the data is returned in CICS-key DSA (CDSA or ECDSA).

SYSID(*systemname*)

(remote and shared queues only) specifies the system name (1–4 characters) identifying the remote system or shared queue pool to which the request is directed.

Conditions**INVREQ**

occurs in either of the following situations:

- the queue was created by CICS internal code.
- the queue name specified consists solely of binary zeroes.

Default action: terminate the task abnormally.

IOERR

occurs when there is an irrecoverable input/output error.

Default action: terminate the task abnormally.

ISCINVREQ

occurs when the remote system indicates a failure that does not correspond to a known condition.

Default action: terminate the task abnormally.

ITEMERR

occurs in any of the following situations:

- The item number specified is invalid (that is, outside the range of item numbers written to the queue).
- An attempt is made to read beyond the end of the queue using the NEXT (default) option.

Default action: terminate the task abnormally.

LENGERR

occurs when the length of the stored data is greater than the value specified by the LENGTH option.

This condition only applies to the INTO option and cannot occur with SET.

Default action: terminate the task abnormally.

NOTAUTH

occurs when a resource security check has failed on QUEUE(*name*).

Default action: terminate the task abnormally.

QIDERR

occurs when the queue specified cannot be found, either in main or in auxiliary storage.

Default action: terminate the task abnormally.

SYSIDERR

occurs in any of the following situations:

- When the SYSID option specifies a name that is neither the local system nor a remote system (made known to CICS by defining a CONNECTION).
- When the link to the remote system is closed.

Default action: terminate the task abnormally.

Examples

The following example shows how to read the first (or only) record from a temporary storage queue into a data area specified in the request; the LENGTH data area is given the value of the length of the record.

```
EXEC CICS READQ TS
      ITEM(1)
      QUEUE(UNIQNAME)
      INTO(DATA)
      LENGTH(LDATA)
```

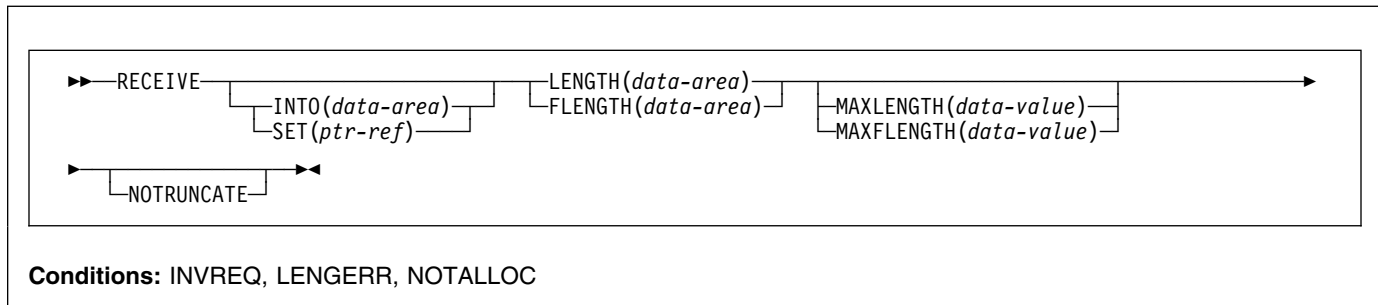
The following example shows how to read the next record from a temporary storage queue into a data area provided by CICS; the pointer reference specified by the SET option is set to the address of the storage area reserved for the data record, and the LENGTH data area is given the value of the length of the record.

```
EXEC CICS READQ TS
      QUEUE(DESCRQ)
      SET(PREF)
      LENGTH(LENG)
      NEXT
```

RECEIVE (VTAM)

RECEIVE (VTAM default)

Receive data from standard CICS terminal support or from a task that is not attached to a terminal.



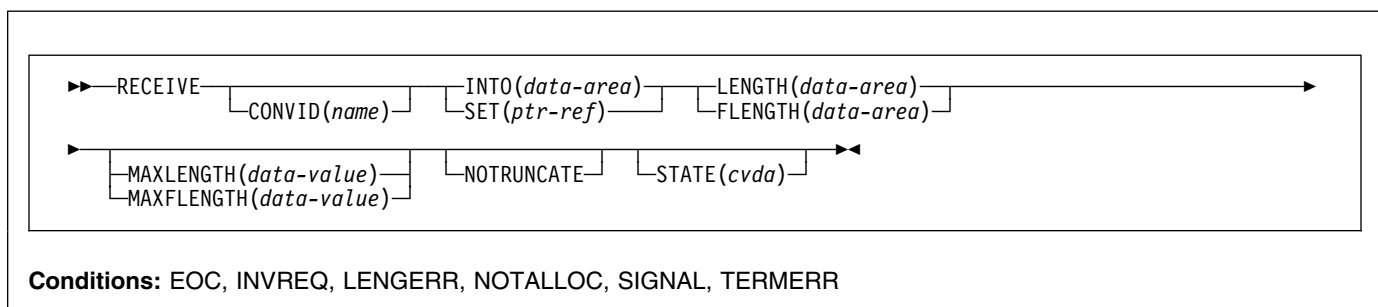
Description

This form of the RECEIVE command is used by all CICS-supported terminals for which the other RECEIVE descriptions are not appropriate.

If data is to be received, you must specify either the INTO or the SET option. If a RECEIVE is issued purely to detect an attention identifier (AID) you can omit both the INTO and SET options.

RECEIVE (APPC)

Receive data on an APPC mapped conversation.

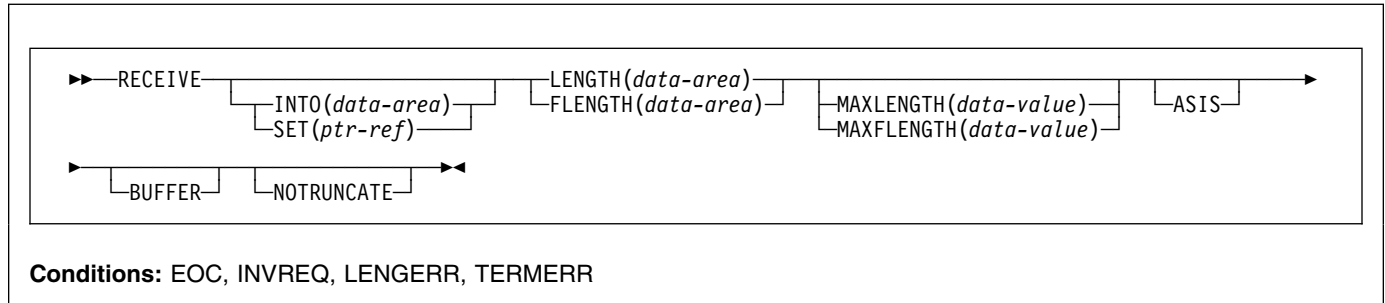


Description

RECEIVE receives data from the conversation partner in an APPC mapped conversation.

RECEIVE (LUTYPE2/LUTYPE3)

Receive data from a 3270-display logical unit (LUTYPE2) or a 3270-printer logical unit (LUTYPE3).



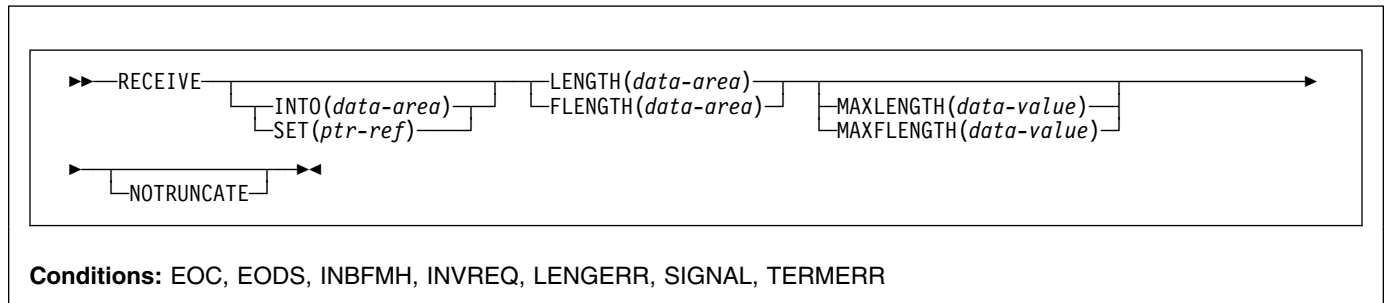
Description

RECEIVE receives data from the terminal.

If data is to be received, you must specify either the INTO or the SET option. If a RECEIVE is issued purely to detect an attention identifier (AID) (and BUFFER has not been specified), you can omit both the INTO and SET options.

RECEIVE (LUTYPE4)

Receive data from an LUTYPE4 logical unit.



Description

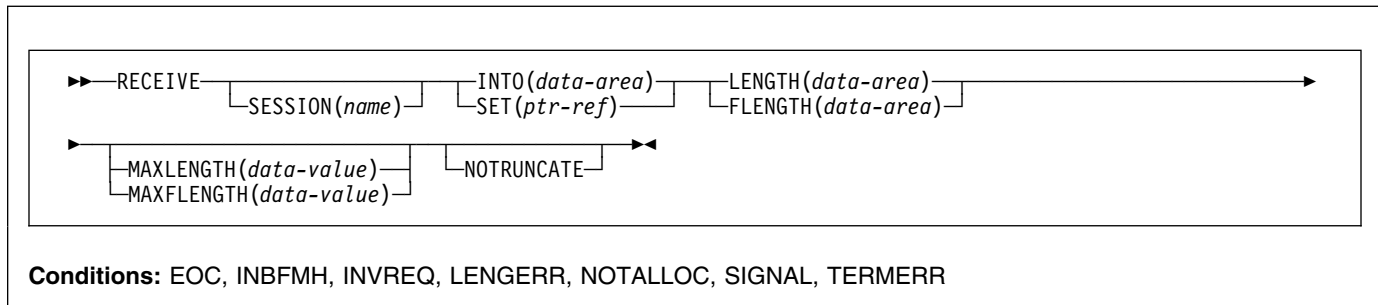
RECEIVE receives data from the terminal.

If data is to be received, you must specify either the INTO or the SET option. If a RECEIVE is issued purely to detect an attention identifier (AID) you can omit both the INTO and SET options.

RECEIVE (VTAM)

RECEIVE (LUTYPE6.1)

Receive data on an LUTYPE6.1 session.

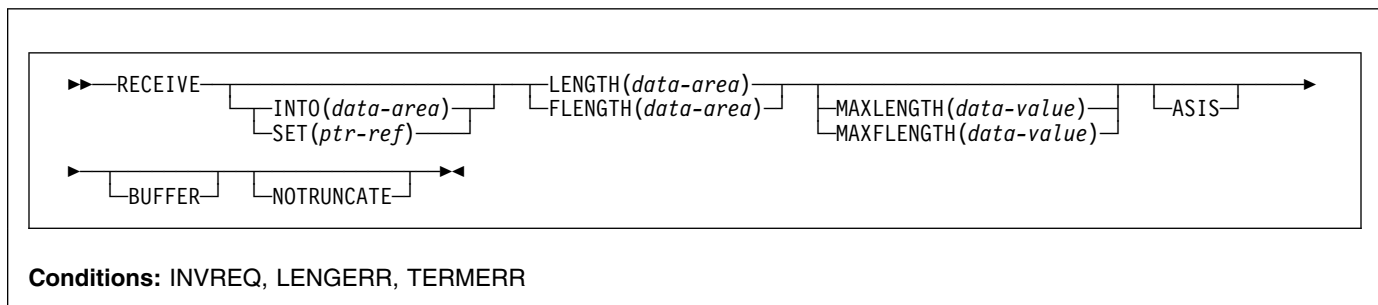


Description

RECEIVE receives data from the conversation partner in an LUTYPE6.1 conversation.

RECEIVE (3270 logical)

Receive data from a 3270 logical unit.



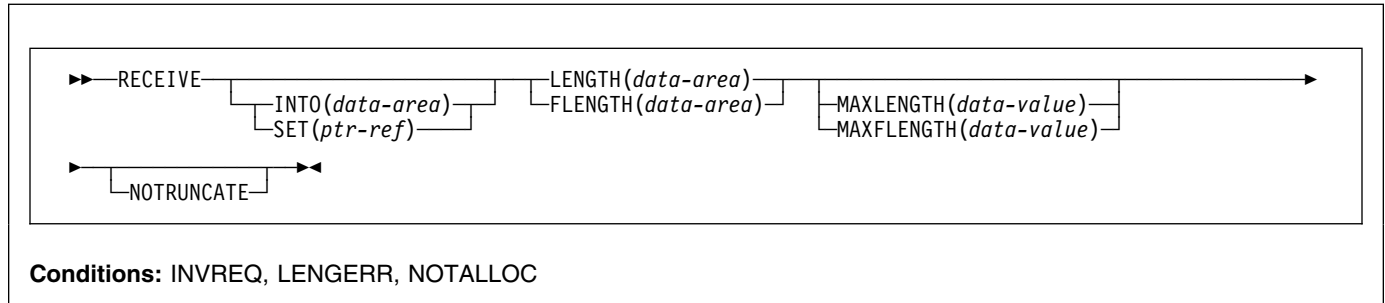
Description

RECEIVE receives data from a terminal.

If data is to be received, you must specify either the INTO or the SET option. If a RECEIVE is issued purely to detect an attention identifier (AID) (and BUFFER has not been specified), you can omit both the INTO and SET options.

RECEIVE (3600 pipeline)

Receive initial input data from a 3600 pipeline logical unit. Subsequent RECEIVES for further input data are not allowed.

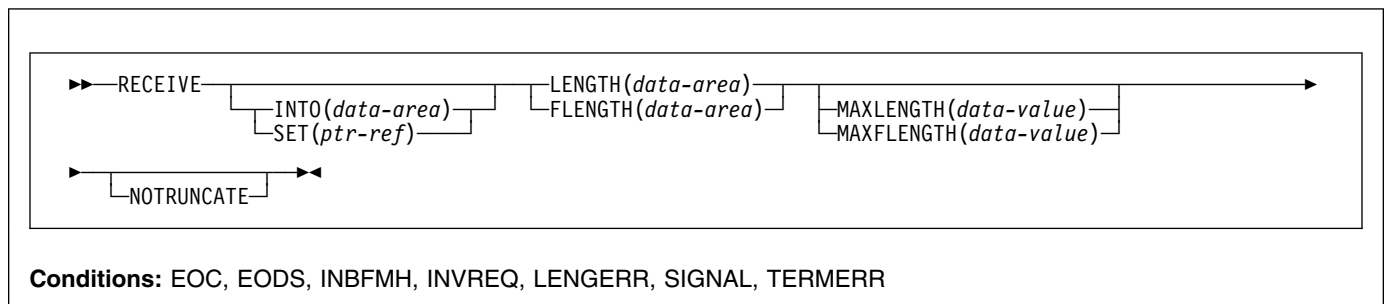


Description

If data is to be received, you must specify either the INTO or the SET option. If a RECEIVE is issued purely to detect an attention identifier (AID) you can omit both the INTO and SET options.

RECEIVE (3600-3601)

Receive data from a 3600 (3601) logical unit.



Description

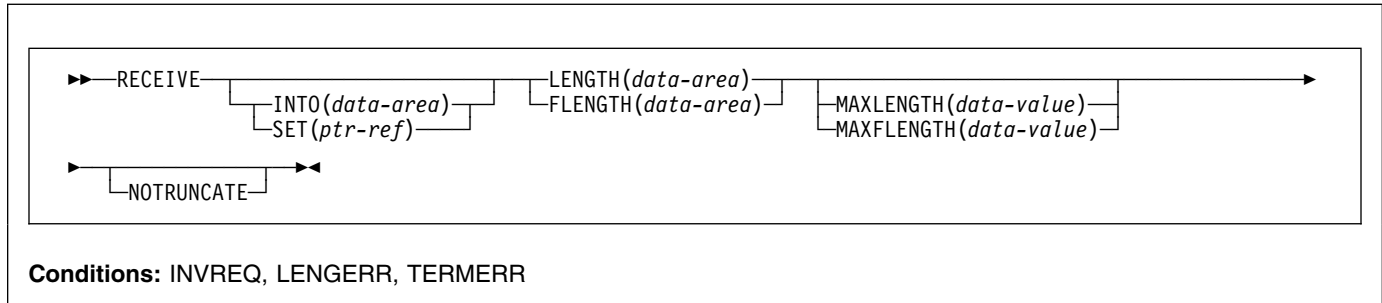
RECEIVE receives data from the terminal. This form of RECEIVE also applies to the 3630 plant communication system.

If data is to be received, you must specify either the INTO or the SET option. If a RECEIVE is issued purely to detect an attention identifier (AID) you can omit both the INTO and SET options.

RECEIVE (VTAM)

RECEIVE (3600-3614)

Receive data from a 3600 (3614) logical unit.



Description

RECEIVE receives data from the terminal.

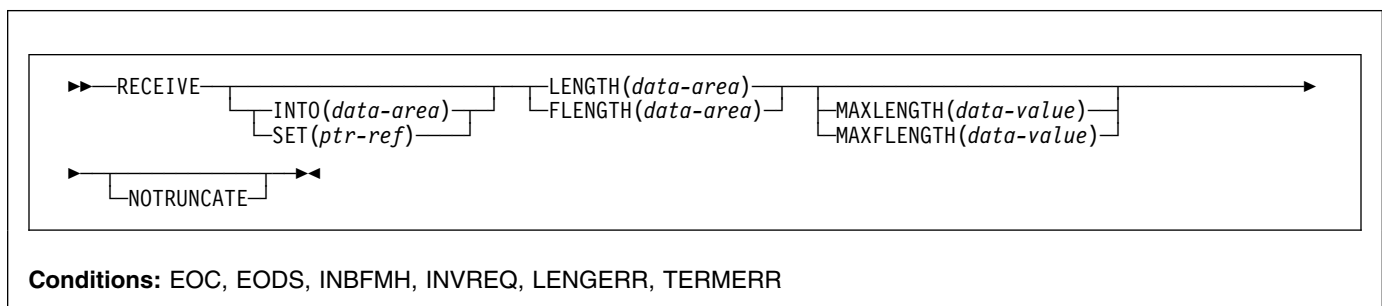
The data-stream and communication format used between a CICS application program and a 3614 is determined by the 3614. The application program is therefore device dependent when handling 3614 communication.

For further information about designing 3614 application programs for CICS, refer to the *CICS/DOS/VS IBM 4700/3600/3630 Guide*.

If data is to be received, you must specify either the INTO or the SET option. If a RECEIVE is issued purely to detect an attention identifier (AID) you can omit both the INTO and SET options.

RECEIVE (3650)

Receive data from 3650 logical units.



Description

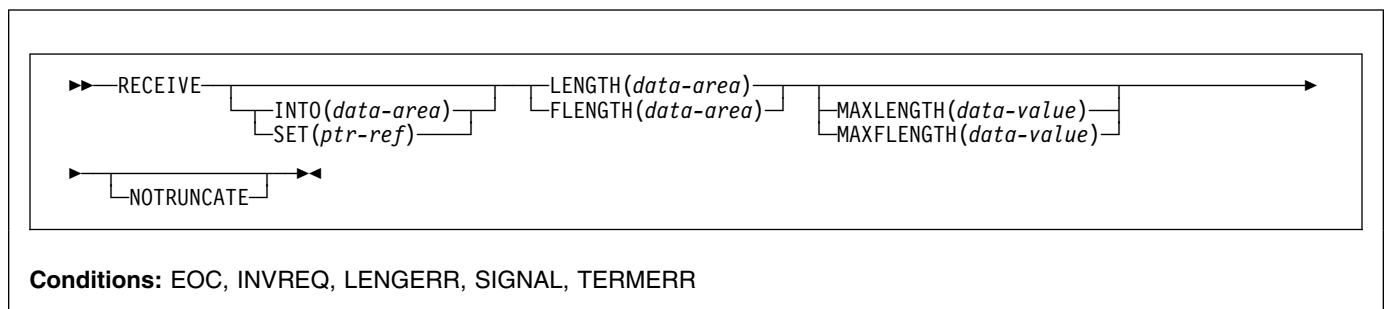
RECEIVE receives data from the terminal. This form of RECEIVE also applies to the following 3650 devices:

- Interpreter logical unit
- Host conversational (3270) logical unit
- Host conversational (3653) logical unit
- 3650/3680 command processor logical unit

If data is to be received, you must specify either the INTO or the SET option. If a RECEIVE is issued purely to detect an attention identifier (AID) you can omit both the INTO and SET options.

RECEIVE (3767)

Receive data from a 3767 interactive logical unit.



Description

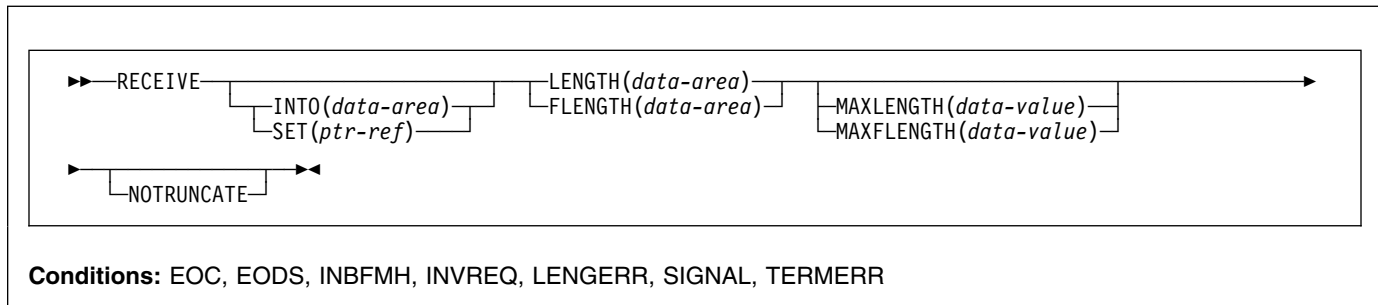
RECEIVE receives data from the terminal. This form of RECEIVE also applies to the 3770 interactive logical unit.

If data is to be received, you must specify either the INTO or the SET option. If a RECEIVE is issued purely to detect an attention identifier (AID) you can omit both the INTO and SET options.

RECEIVE (VTAM)

RECEIVE (3770)

Receive data from a 3770 batch logical unit.



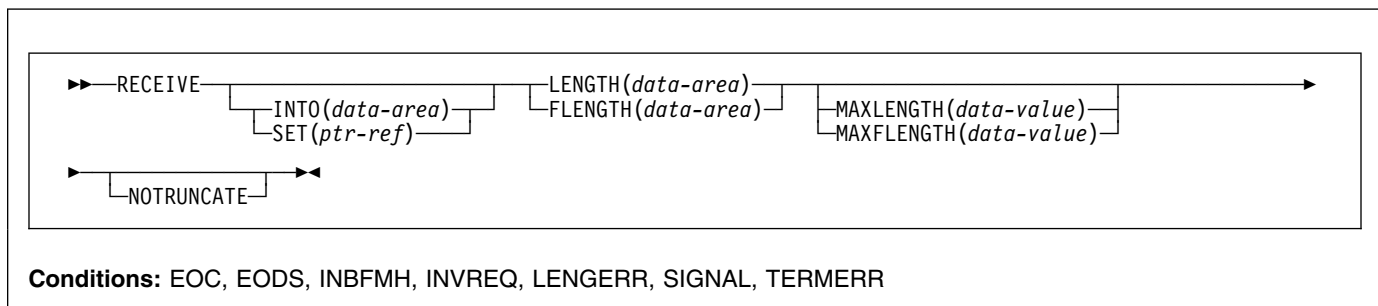
Description

RECEIVE receives data from the terminal.

If data is to be received, you must specify either the INTO or the SET option. If a RECEIVE is issued purely to detect an attention identifier (AID) you can omit both the INTO and SET options.

RECEIVE (3790 full-function or inquiry)

Receive data from a 3790 full-function or inquiry logical unit.



Description

RECEIVE receives data from the terminal. This form of RECEIVE also applies to the following devices:

- 3650/3680 full-function logical unit
- 3770 full-function logical unit

If data is to be received, you must specify either the INTO or the SET option. If a RECEIVE is issued purely to detect an attention identifier (AID) you can omit both the INTO and SET options.

Options

ASIS

specifies that lowercase characters in the 3270 input data stream are not translated to uppercase; this allows the current task to receive a message containing both uppercase and lowercase data.

This option has no effect on the first RECEIVE command of a transaction, because terminal control performs a READ INITIAL and uses the terminal defaults to translate the operation data.

This option has no effect if the screen contains data prior to a transaction being initiated. This data is read and translated in preparation for the next task and the first RECEIVE command in that task retrieves the translated data.

Note: If you are using a katakana terminal, you might see some messages containing mixed English and katakana characters. That is because katakana terminals cannot display mixed-case output. Uppercase characters in the data stream appear as uppercase English characters, but lowercase characters appear as katakana characters. If this happens, ask your system programmer to specify MSGCASE=UPPER in the system initialization parameters so that messages contain uppercase characters only.

BUFFER

specifies that the contents of the 3270 buffer are to be read, beginning at buffer location 1 and continuing until all contents of the buffer have been read. All character and attribute sequences (including nulls) appear in the input data stream in the same order as they appear in the 3270 buffer.

CONVID(*name*)

identifies the conversation to which the command relates. The 4-character name identifies either the token returned by a previously executed ALLOCATE command in EIBRSRCE in the EIB, or the token representing the principal session (returned by a previously executed ASSIGN command).

For compatibility with earlier releases, SESSION is accepted as a synonym for CONVID. New programs should use CONVID.

If this option is omitted, the principal facility is assumed.

FLENGTH(*data-area*)

A fullword alternative to LENGTH.

INTO(*data-area*)

specifies the receiving field for the data read from the logical unit or terminal, or the application target data area into which data is to be received from the application program connected to the other end of the current conversation.

LENGTH(*data-area*)

specifies the length, as a halfword binary value, of the data received.

If you specify the INTO option, but omit the MAXLENGTH option, the argument must be a data area that specifies the maximum length that the program accepts. If the value specified is less than zero, zero is assumed.

If you specify the SET option, the argument must be a data area. When the data has been received, the data area is set to the length of the data.

MAXFLENGTH(*data-value*)

A fullword alternative to MAXLENGTH.

MAXLENGTH(*data-value*)

specifies the maximum amount (halfword binary value) of data that CICS is to recover. If INTO is specified, MAXLENGTH overrides the use of LENGTH as an input to CICS. If SET is specified, MAXLENGTH provides a way for the program to limit the amount of data it receives at one time.

If the length of data exceeds the value specified and the NOTRUNCATE option is not present, the data is truncated to that value and the LENGERR condition occurs. The data area specified in the LENGTH option is set to the original length of data.

If the length of data exceeds the value specified and the NOTRUNCATE option is present, CICS retains the remaining data and uses it to satisfy subsequent RECEIVE commands. The data area specified in the LENGTH option is set to the length of data returned.

If this option is omitted, the value indicated in the LENGTH option is assumed.

NOTRUNCATE

specifies that, when the data available exceeds the length requested, the remaining data is not to be discarded but is to be retained for retrieval by subsequent RECEIVE commands.

SESSION(*name*)

specifies the symbolic identifier (1–4 characters) of a session TCTTE. This option specifies the alternate facility to be used. If this option is omitted, the principal facility for the task is used.

SET(*ptr-ref*)

specifies the pointer reference that is to be set to the address of the data read from the logical unit or terminal, or the partner transaction. The pointer reference is valid until the next receive command or the end of task.

If DATALOCATION(ANY) is specified in the RDO PROGRAM resource definition for the application program, the address of the data can be above or below the 16MB line.

If DATALOCATION(BELOW) is specified in the RDO PROGRAM resource definition for the application program, and the data resides above the 16MB line, the data is copied below the 16MB line, and the address of this copy is returned.

RECEIVE (VTAM)

If TASKDATAKEY(USER) is specified in the RDO TRANSACTION resource definition for the executing transaction, and storage protection is active, the data returned is in user-key DSA (UDSA or EUDSA); otherwise the data is returned in CICS-key DSA (CDSA or ECDSA).

STATE(*cvda*)

gets the state of the current conversation. The *cvda* values returned by CICS are:

ALLOCATED
CONFFREE
CONFRECEIVE
CONFSEND
FREE
PENDFREE
PENDRECEIVE
RECEIVE
ROLLBACK
SEND
SYNCFREE
SYNCRECEIVE
SYNCSEND

Conditions

Some of the following conditions may occur in combination with others. CICS checks for these conditions in the following order:

1. EODS
2. INBFMH
3. EOC

If more than one occurs, only the first is passed to the application program. EIBRCODE, however, is set to indicate all the conditions that occurred.

EOC

occurs when a request/response unit (RU) is received with end-of-chain-indicator set. Field EIBEOC also indicates this condition.

Default action: ignore the condition.

EODS (interpreter logical unit only)

occurs when an end-of-data-set indicator is received.

Default action: terminate the task abnormally.

INBFMH

occurs if a request/response unit (RU) contains a function management header (FMH). Field EIBFMH contains this indicator and it should be used in preference to INBFMH. The IGNORE CONDITION command can be used to ignore the condition.

Default action: terminate the task abnormally.

INVREQ

RESP2 values:

200 A distributed program link server application specified the function-shipping session (its principal facility) on the CONVID option.

also occurs (RESP2 not set) in any of the following situations:

- The command is used on an APPC conversation that is not using the EXEC CICS interface or that is not a mapped conversation.

Default action: terminate the task abnormally.

LENGERR

occurs if data is discarded by CICS because its length exceeds the maximum the program accepts and the NOTRUNCATE option is not specified.

Default action: terminate the task abnormally.

NOTALLOC

occurs if the RECEIVE command is issued by a transaction that has been started as a nonterminal task by the START command, or if the CONVID value or facility specified in the command does not relate to a conversation owned by the application.

Default action: terminate the task abnormally.

SIGNAL

occurs when an inbound SIGNAL data-flow control command is received from a partner transaction. EIBSIG is always set when an inbound signal is received.

Default action: ignore the condition.

TERMERR

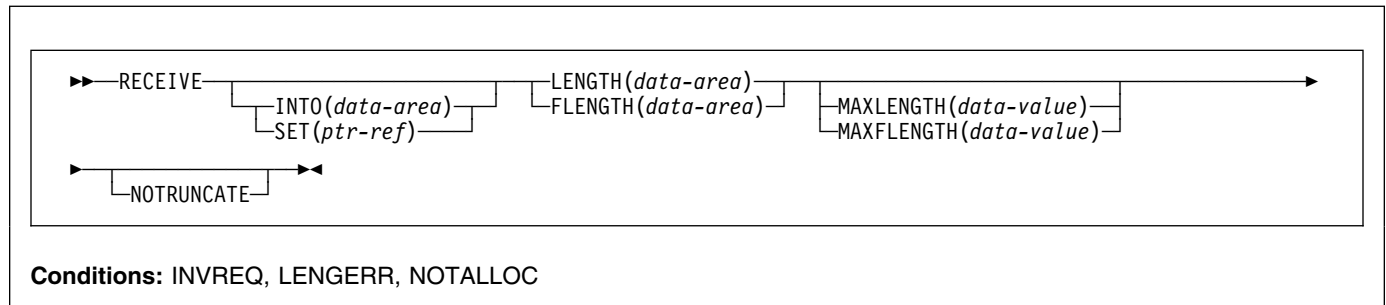
occurs for a session-related or terminal-related error. Any action on that conversation other than a FREE causes an ATCV abend.

A CANCEL TASK request by a user node error program (NEP) may cause this condition if the task has an outstanding terminal control request active when the node abnormal condition program handles the session error.

Default action: terminate the task abnormally with abend code ATNI.

RECEIVE (non-VTAM default)

Receive data from standard CICS terminal support or from a task that is not attached to a terminal.



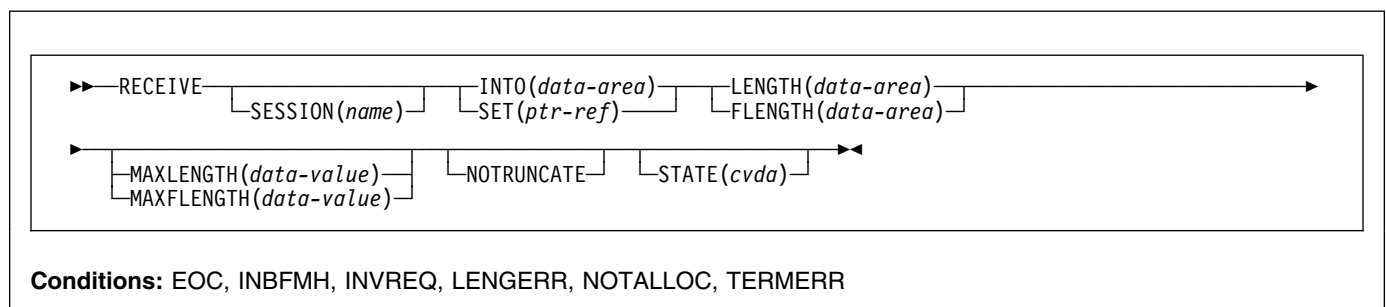
Description

This form of the RECEIVE command is used by all CICS-supported terminals for which the other RECEIVE descriptions are not appropriate.

If data is to be received, you must specify either the INTO or the SET option. If a RECEIVE is issued purely to detect an attention identifier (AID) you can omit both the INTO and SET options.

RECEIVE (MRO)

Receive data on an MRO conversation.



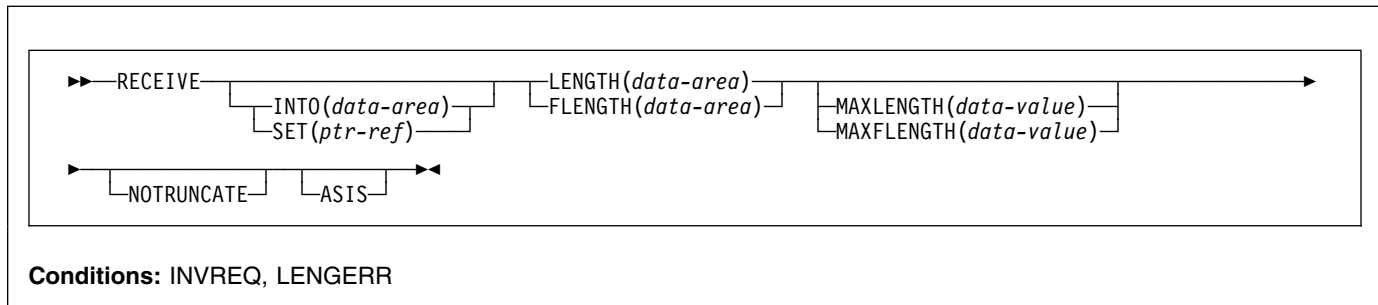
Description

RECEIVE receives data from the conversation partner in an MRO conversation.

RECEIVE (non-VTAM)

RECEIVE (System/3)

Receive data from a System/3.



Description

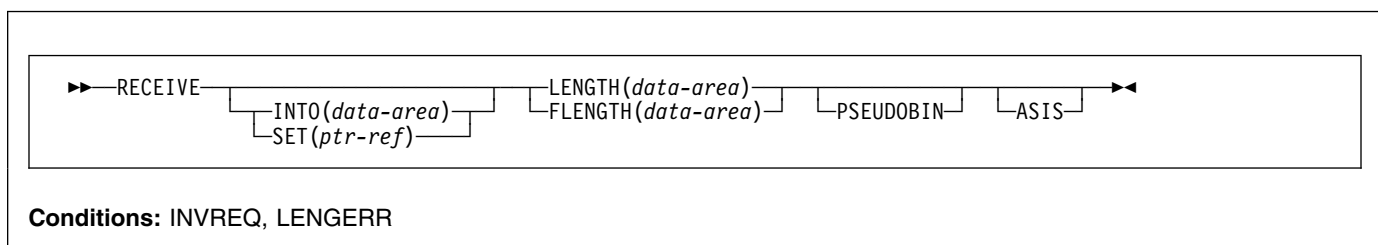
RECEIVE receives data from the terminal. This form of the RECEIVE command also applies to the following devices:

- 2770 data communication system
- 2780 data transmission terminal
- 3660 supermarket scanning system
- 3780 communication terminal

If data is to be received, you must specify either the INTO or the SET option. If a RECEIVE is issued purely to detect an attention identifier (AID) you can omit both the INTO and SET options.

RECEIVE (System/7)

Receive data from a System/7.



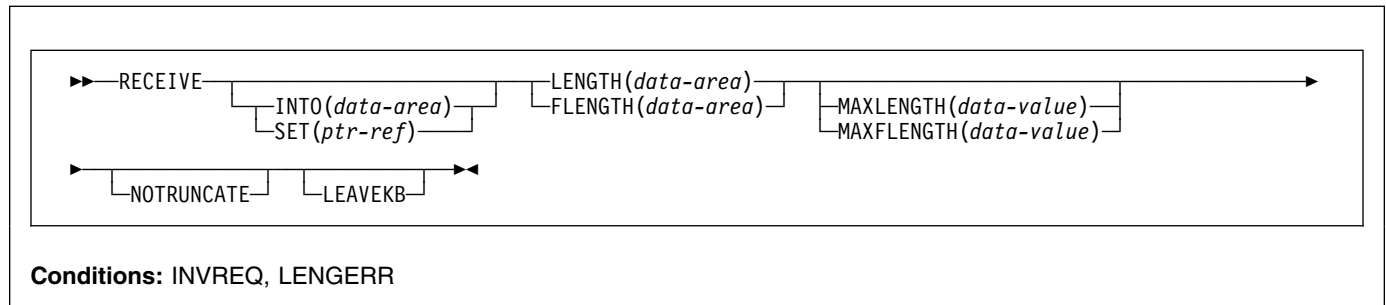
Description

RECEIVE receives data from the terminal. For a description of running with a System/7, see "CONVERSE (System/7)" on page 53.

If data is to be received, you must specify either the INTO or the SET option. If a RECEIVE is issued purely to detect an attention identifier (AID) you can omit both the INTO and SET options.

RECEIVE (2260)

Receive data from a 2260 or 2265 display station.



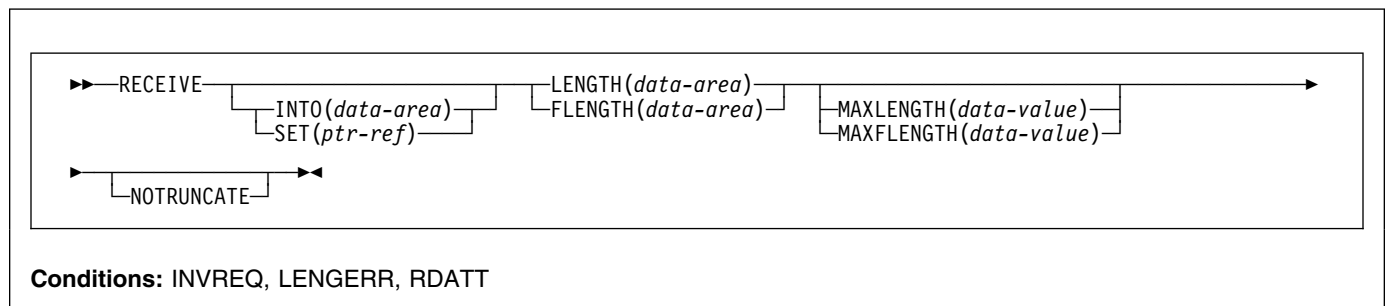
Description

RECEIVE receives data from the terminal.

If data is to be received, you must specify either the INTO or the SET option. If a RECEIVE is issued purely to detect an attention identifier (AID) you can omit both the INTO and SET options.

RECEIVE (2741)

Receive data from a 2741 communication terminal.



Description

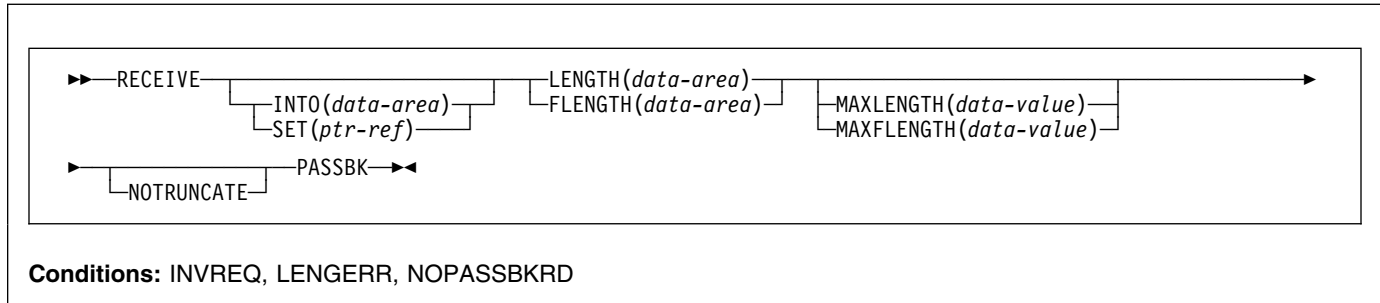
RECEIVE receives data from the terminal. For more information about using a 2741 communication terminal, see "CONVERSE (2741)" on page 55.

If data is to be received, you must specify either the INTO or the SET option. If a RECEIVE is issued purely to detect an attention identifier (AID) you can omit both the INTO and SET options.

RECEIVE (non-VTAM)

RECEIVE (2980)

Receive data from a 2980 general banking terminal system.



Description

RECEIVE receives data from the terminal.

If data is to be received, you must specify either the INTO or the SET option. If a RECEIVE is issued purely to detect an attention identifier (AID) you can omit both the INTO and SET options.

Passbook control

All input and output requests to the passbook area of a 2980 are dependent on the presence of a passbook. The PASSBK option is used to specify that communication is with a passbook. The conditions NOPASSBKRD (RECEIVE) and NOPASSBKWR (SEND) occur on input and output requests respectively when a passbook is not present. These conditions can be handled by a HANDLE CONDITION command and appropriate handling routines.

If the passbook is present on an input request, the application program generally writes back to the passbook area to update the passbook. If the NOPASSBKWR condition occurs, CICS allows immediate output to the terminal. In a routine for the NOPASSBKWR condition, the application program should send an error message to the journal area of the terminal to inform the 2980 operator of this error condition. To allow the operator to insert the required passbook, CICS causes the transaction to wait 23.5 seconds before continuing.

On regaining control from CICS after sending the error message, the application program can attempt again to update the passbook when it has ensured that the print element is positioned correctly in the passbook area. This is generally accomplished by issuing two carrier returns followed by the number of tabs required to move the print element to the correct position.

If the NOPASSBKWR condition occurs during the second attempt to write to the passbook area, the application program can send another error message or take some alternative action (for example, place the terminal "out of service"). The presence of the Auditor Key on a 2980 Administrative Station Model 2 is controlled by the SEND PASSBK command and may be used in a manner similar to that described above.

Output control

The unit of transmission for a 2980 is called a **segment**. However, for the passbook and journal areas, CICS allows an application program to send messages that exceed the buffer size. For the passbook area, the maximum length of message is limited to one line of a passbook to avoid spacing ("indexing") past the bottom of the passbook. For the journal area, the maximum length of message is specified in the LENGTH option of the SEND command.

For example, consider a 2972 buffer size of 48 characters and a 2980 Teller Station Model 4 passbook print area of 100 characters per line. The application program can send a message of 100 characters to this area; CICS segments the message to adjust to the buffer size. The application program must insert the passbook indexing character (X'25') as the **last** character

written in one output request to the passbook area. This is done to control passbook indexing and thereby achieve positive control of passbook presence.

If a message contains embedded passbook indexing characters, and segmentation is necessary because of the length of the message, the output is terminated if the passbook spaces beyond the bottom of the passbook; the remaining segments are not printed.

Output to a common buffer

The SEND CBUFF command is used to transmit data to a common buffer. The data is translated to the character set of the receiving 2980 model. If more than one 2980 model type is connected to the 2972 control unit, the lengths are truncated if they exceed the buffer size.

The DFH2980 structure

The DFH2980 structure contains constants that may only be used when writing COBOL or PL/I application programs for the 2980. The structure is obtained by copying DFH2980 into the application program.

For COBOL, DFH2980 is copied into the working-storage section; for PL/I, DFH2980 is included using a %INCLUDE statement.

The station identification is given in the field STATIONID, whose value must be determined by the ASSIGN command. To test whether a normal or alternate station is being used, the STATIONID field is compared with values predefined in DFH2980. The values are:

STATION-
STATION_

where normal stations. (The break symbol is hyphen (-) for COBOL, and underscore (_) for PL/I.)

The teller identification on a 2980 Teller Station Model 4 is given in the 1-byte character field TELLERID. An ASSIGN command must be used to find out the TELLERID value.

Tab characters (X'05') must be included in the application program. The number of tabs required to position the print element to the first position of a passbook area is given in the field NUMTAB. An ASSIGN command must be used to find out the NUMTAB value. The value of NUMTAB is specified by the system programmer and may be unique to each terminal.

Other tab characters are inserted as needed to control formatting.

Any of the DFH2980 values TAB-ZERO through TAB-NINE for COBOL and PL/I, may be compared with NUMTAB to find out the number of tab characters that need to be inserted in an output message to get correct positioning of the print element. The tab character is included in DFH2980 as TABCHAR.

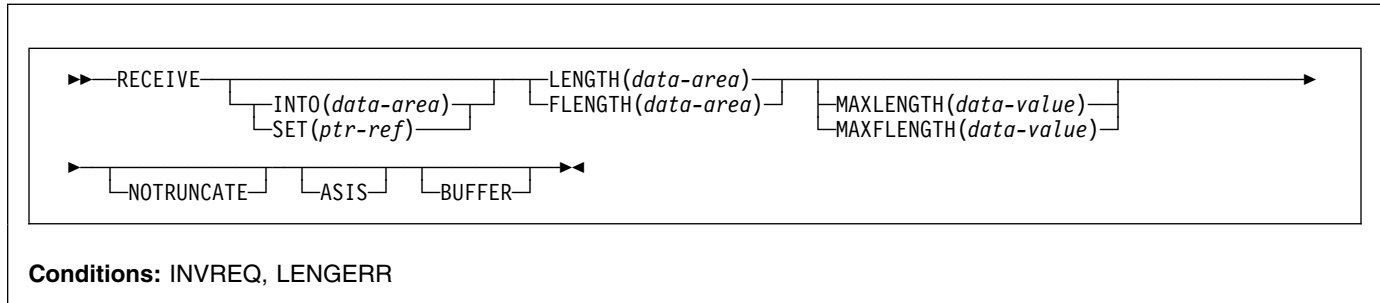
Thirty special characters are defined in DFH2980. Twenty-three of these can be referred to by the name SPECCHAR-# or SPECCHAR_# (for American National Standard COBOL or PL/I) where # is an integer (0 through 22). The seven other characters are defined with names that imply their usage, for example, TABCHAR.

Several other characters defined in DFH2980, such as HOLDPCF or TCTTEPCR, are intended for use in application programs using CICS macros, and should not be required in application programs using CICS commands.

RECEIVE (non-VTAM)

RECEIVE (3270 display)

Receive data from a 3270 information display system.



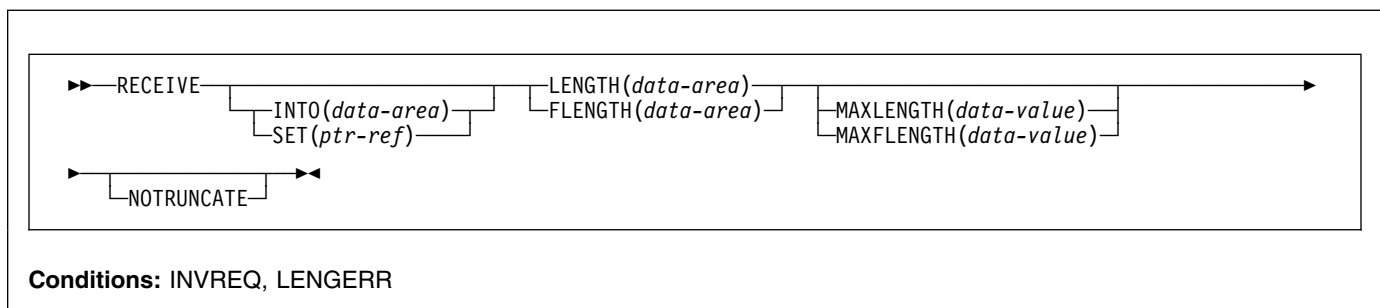
Description

RECEIVE receives data from the terminal.

If data is to be received, you must specify either the INTO or the SET option. If a RECEIVE is issued purely to detect an attention identifier (AID) (and BUFFER has not been specified), you can omit both the INTO and SET options.

RECEIVE (3600 BTAM)

Receive data from a 3600 finance communication system (BTAM) - only valid for remote terminals. This form of the RECEIVE command also applies to the 4700 finance communication system. For more information about using the 3600 or 4700 finance communication system, see "CONVERSE (3600 BTAM)" on page 57 and "BTAM programmable terminals" on page 388.



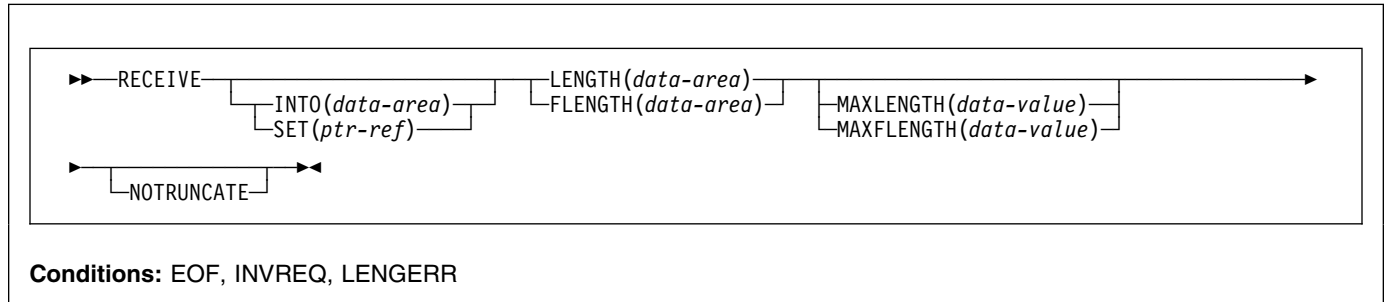
Description

RECEIVE receives data from the terminal. For more information about using the 3600 or 4700 finance communication system, see "CONVERSE (3600 BTAM)" on page 57.

If data is to be received, you must specify either the INTO or the SET option. If a RECEIVE is issued purely to detect an attention identifier (AID) you can omit both the INTO and SET options.

RECEIVE (3735)

Receive data from a 3735 Programmable Buffered Terminal.



Description

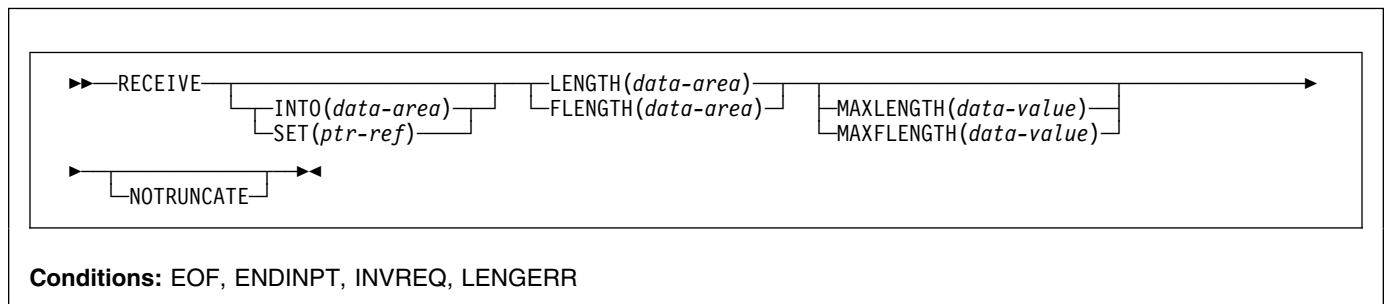
RECEIVE receives data from the terminal. The 3735 Programmable Buffered Terminal may be serviced by CICS in response to terminal-initiated input (automatic answering), or as a result of an automatic (automatic calling) or time-initiated transaction.

For more information about using the 3735 terminal, see “CONVERSE (3735)” on page 59.

If data is to be received, you must specify either the INTO or the SET option. If a RECEIVE is issued purely to detect an attention identifier (AID) you can omit both the INTO and SET options.

RECEIVE (3740)

Receive data from a 3740 data entry system.



Description

RECEIVE receives data from the terminal.

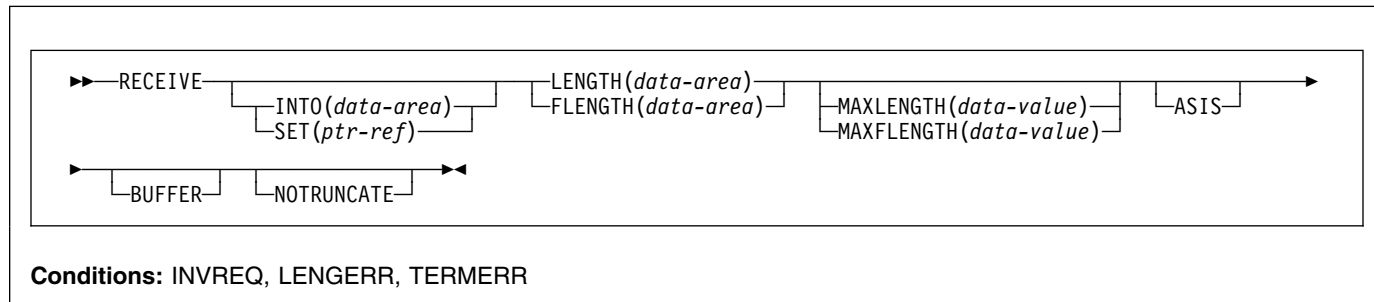
In batch mode, many files are exchanged between the 3740 and CICS in a single transmission. The transmission of an input batch must be complete before an output transmission can be started.

If data is to be received, you must specify either the INTO or the SET option. If a RECEIVE is issued purely to detect an attention identifier (AID) you can omit both the INTO and SET options.

RECEIVE (non-VTAM)

RECEIVE (3790 3270-display)

Receive data from a 3790 (3270-display) logical unit.



Description

RECEIVE receives data from the terminal.

If data is to be received, you must specify either the INTO or the SET option. If a RECEIVE is issued purely to detect an attention identifier (AID) (and BUFFER has not been specified), you can omit both the INTO and SET options.

Options

If you specify the INTO option, but omit the MAXLENGTH option, the argument must be a data area that specifies the maximum length that the program accepts. If the value specified is less than zero, zero is assumed.

If the length of the data exceeds the value specified, but the NOTRUNCATE option is not specified, the data is truncated to that value and the LENGERR condition occurs. When the data has been received, the data area is set to the original length of the data.

If you specify the SET option, the argument must be a data area. When the data has been received, the data area is set to the length of the data.

For a description of a safe upper limit, see “LENGTH options” on page 5.

ASIS

indicates that output is to be sent in transparent mode (with no recognition of control characters and accepting any of the 256 possible combinations of eight bits as valid transmittable data).

This option has no effect on the first RECEIVE command of a transaction, because terminal control performs a READ INITIAL operation and uses the terminal defaults to translate the data.

This option has no effect if the screen contains data prior to a transaction being initiated. This data is read and translated in preparation for the next task and the first RECEIVE command in that task retrieves the translated data.

Note: If you are using a katakana terminal, you might see some messages containing mixed English and katakana characters. That is because katakana terminals cannot display mixed-case output. Uppercase characters in the data stream appear as uppercase English characters, but lowercase characters appear as katakana characters. If this happens, ask your system programmer to specify MSGCASE=UPPER in the system initialization parameters so that messages contain uppercase characters only.

BUFFER

specifies that the contents of the 3270 buffer are to be read, beginning at buffer location one and continuing until all contents of the buffer have been read. All character and attribute sequences (including nulls) appear in the input data stream in the same order that they appear in the 3270 buffer.

FLENGTH(*data-area*)

A fullword alternative to LENGTH.

INTO(*data-area*)

specifies the receiving field for the data read from the terminal or logical unit, or the application target area receiving the data from the application program connected to the other end of the current conversation.

LEAVEKB

specifies that the keyboard is to remain locked at the completion of the data transfer.

LENGTH(*data-area*)

specifies the length, as a halfword binary value, of the data transmitted.

If you specify the INTO option, but omit the MAXLENGTH option, the argument must be a data area that specifies the maximum length that the program accepts. If the value specified is less than zero, zero is assumed.

If the length of the data exceeds the value specified, but the NOTRUNCATE option is not specified, the data is truncated to that value and the LENGERR condition occurs. When the data has been received, the data area is set to the original length of the data.

If you specify the SET option, the argument must be a data area. When the data has been received, the data area is set to the length of the data.

For a description of a safe upper limit, see “LENGTH options” on page 5.

MAXLENGTH(*data-value*)

A fullword alternative to MAXLENGTH.

MAXLENGTH(*data-value*)

specifies the maximum amount (halfword binary value) of data that CICS is to recover. If INTO is specified, MAXLENGTH overrides the use of LENGTH as an input to CICS. If SET is specified, MAXLENGTH provides a way for the program to limit the amount of data it receives at one time.

If the value specified is less than zero, zero is assumed.

If the length of data exceeds the value specified and the NOTRUNCATE option is not present, the data is truncated to that value and the LENGERR condition occurs. The data area specified in the LENGTH option is set to the original length of data.

If the length of data exceeds the value specified and the NOTRUNCATE option is present, CICS retains the remaining data and uses it to satisfy subsequent RECEIVE commands. The data area specified in the LENGTH option is set to the length of data returned.

If this option is omitted, the value indicated in the LENGTH option is assumed.

NOTRUNCATE

specifies that, when the data available exceeds the length requested, the remaining data is not to be discarded but is to be retained for retrieval by subsequent RECEIVE commands.

PASSBK

specifies that communication is with a passbook.

RECEIVE (non-VTAM)

PSEUDOBIN

specifies that the data being read is to be translated from System/7 pseudobinary representation to hexadecimal.

SESSION(*name*)

specifies the symbolic identifier (1–4 characters) of a session TCTTE. This option specifies the alternate facility to be used. If this option is omitted, the principal facility for the task is used.

SET(*ptr-ref*)

specifies a pointer reference that is to be set to the address of data received from the conversation partner in an MRO conversation. The pointer reference is valid until the next receive command or the end of task.

If DATALOCATION(ANY) is specified in the RDO PROGRAM resource definition for the application program, the address of the data can be above or below the 16MB line.

If DATALOCATION(BELOW) is specified in the RDO PROGRAM resource definition for the application program, and the data resides above the 16MB line, the data is copied below the 16MB line, and the address of this copy is returned.

If TASKDATAKEY(USER) is specified in the RDO TRANSACTION resource definition for the executing transaction, and storage protection is active, the data returned is in user-key DSA (UDSA or EUDSA); otherwise the data is returned in CICS-key DSA (CDSA or ECDSA).

STATE(*cvda*)

gets the state of the current conversation. The *cvda* values returned by CICS are:

ALLOCATED
FREE
PENDFREE
RECEIVE
ROLLBACK
SEND
SYNCFREE
SYNCRECEIVE
SYNCSSEND

Conditions

The following conditions can occur in combination with others. CICS checks for these conditions in the order:

1. INBFMH
2. EOC

If more than one occurs, only the first is passed to the application program. EIBRCODE, however, is set to indicate all the conditions that occurred.

ENDINPT

occurs when an end-of-input indicator is received.

Default action: terminate the task abnormally.

EOC

occurs when a request/response unit (RU) is received with the end-of-chain indicator set. Field EIBEOC also contains this indicator.

Default action: ignore the condition.

EOF

occurs when an end-of-file indicator is received.

Default action: terminate the task abnormally.

INBFMH

occurs if a request/response unit (RU) contains a function management header (FMH). Field EIBFMH contains this indicator and it should be used in preference to INBFMH. The IGNORE CONDITION command can be used to ignore the condition.

Default action: terminate the task abnormally.

INVREQ

RESP2 values:

200 A distributed program link server application specified the function-shipping session (its principal facility) on the CONVID option.

Default action: terminate the task abnormally.

LENGERR

occurs if data is discarded by CICS because its length exceeds the maximum the program accepts and the NOTRUNCATE option is not specified.

Default action: terminate the task abnormally.

NOPASSBKRD

occurs if no passbook is present.

Default action: terminate the task abnormally.

NOTALLOC

occurs if the facility specified in the command is not owned by the application.

Default action: terminate the task abnormally.

RDATT

occurs if a RECEIVE command is terminated by the attention (ATTN) key rather than the return key.

Default action: ignore the condition.

TERMERR

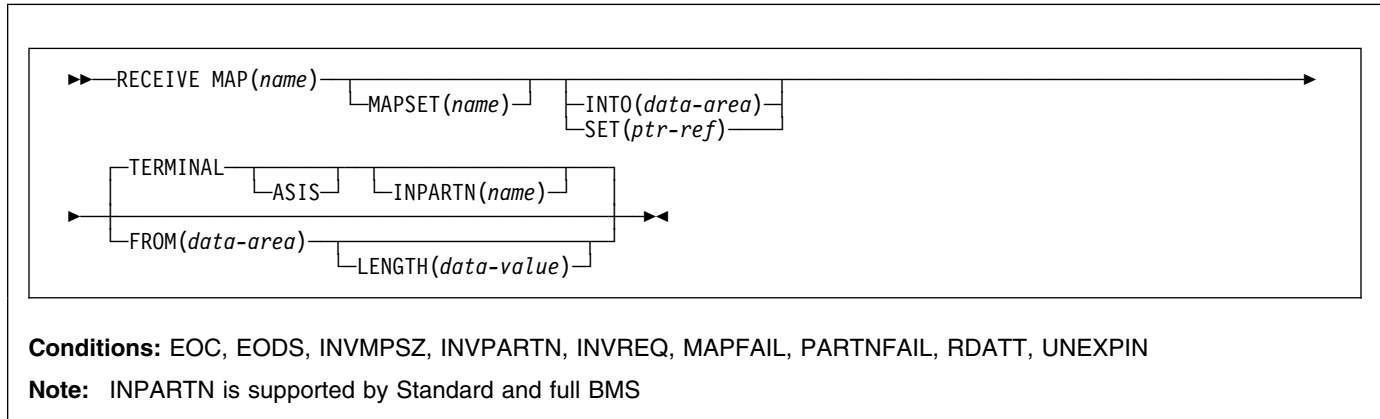
occurs for a terminal-related error, such as a session failure. This condition applies to VTAM-connected terminals only.

A CANCEL TASK request by a user node error program (NEP) may cause this condition if the task has an outstanding terminal control request active when the node abnormal condition program handles the session error.

Default action: terminate the task abnormally with abend code ATNI.

RECEIVE MAP

Receive screen input into an application data area. For further information about BMS, see the *CICS Application Programming Guide*.



Description

RECEIVE MAP maps input data from a terminal into a data area in an application program.

Data from certain logical units is not mapped, but is left unaltered. Refer to the appropriate CICS subsystem guide to see if this is true for a particular logical unit.

Following a RECEIVE MAP command, the inbound cursor position is placed in EIBCPOSN, and the terminal attention identifier (AID) placed in EIBAID.

See Appendix K, "BMS macro summary" on page 411 for map definitions.

If data is to be received, you must specify either the INTO or the SET option. If a RECEIVE is issued purely to detect an attention identifier (AID), you can omit both the INTO and the SET options.

Options

ASIS

specifies that lowercase characters in the 3270 input data stream are not translated to uppercase; this allows the current task to receive a message containing both uppercase and lowercase data.

This option has no effect on the first RECEIVE command of a transaction, or if the screen contains data prior to a transaction being initiated. For example, if a transaction is initiated by another transaction, and begins by receiving data originally output by that transaction, it cannot suppress uppercase translation on the data. This data is read and translated in preparation for the next

task and the first RECEIVE command in that task retrieves the translated data.

Note: If you are using a katakana terminal, you might see some messages containing mixed English and katakana characters. That is because katakana terminals cannot display mixed-case output. Uppercase characters in the data stream appear as uppercase English characters, but lowercase characters appear as katakana characters. If this happens, ask your system programmer to specify MSGCASE=UPPER in the system initialization parameters so that messages contain uppercase characters only.

FROM(data-area)

specifies the data area containing the data to be mapped by a RECEIVE MAP command. This includes the 12-byte prefix generated by the TIOAPFX=YES option on the DFHMDI and DFHMSD BMS map definitions (see pages 425 and 432).

INPARTN(name)

specifies the name (1–2 characters) of the partition in which the terminal operator is expected to enter data. If the terminal operator enters data in some other partition, the INPARTN partition is activated, the keyboard is unlocked for the partition, and an error message is output to any error message partition. This option is ignored if the terminal does not support partitions, or if there is no application partition set.

INTO(data-area)

specifies the data area into which the mapped data is to be written. If this field is not specified, the name defaults to the name of the map suffixed with an I.

LENGTH(data-value)

specifies the length of the data to be formatted as a halfword binary value. It must not exceed the length of

the FROM data area, but this should include the length of the 12-byte prefix generated by the TIOAPFX=YES option on the DFHMDI and DFHMSD BMS map definitions (see pages 425 and 432).

For a description of a safe upper limit, see “LENGTH options” on page 5.

MAP(name)

specifies the name (1–7 characters) of the map to be used.

MAPSET(name)

specifies the unsuffixed name (1–7 characters) of the mapset to be used. The mapset must reside in the CICS program library. The mapset can be defined either by using RDO or by program autoinstall when the mapset is first used. If this option is not specified, the name given in the MAP option is assumed to be that of the mapset.

SET(ptr-ref)

specifies the pointer that is to be set to the address of the 12-byte prefix to the mapped data.

The pointer reference is valid until the next receive command or the end of task.

If DATALOCATION(ANY) is specified in the RDO PROGRAM resource definition for the application program, the address of the data may be above or below the 16MB line.

If DATALOCATION(BELOW) is specified in the RDO PROGRAM resource definition for the application program, and the data resides above the 16MB line, the data is copied below the 16MB line, and the address of this copy is returned.

If TASKDATAKEY(USER) is specified in the RDO TRANSACTION resource definition for the executing transaction, and storage protection is active, the data returned is in user-key DSA (UDSA or EUDSA); otherwise the data is returned in CICS-key DSA (CDSA or ECDSA).

TERMINAL

specifies that input data is to be read from the terminal that originated the transaction.

Conditions

Some of the following conditions can occur in combination. If more than one occurs, only the first is passed to the application program.

EIBRCODE, however, is set to indicate all the conditions that occurred.

EOC

occurs if the request/response unit (RU) is received with the end-of-chain (EOC) indicator set. It applies only to logical units.

Default action: ignore the condition.

EODS

occurs if no data is received (only an FMH). It applies only to 3770 batch LUs and to 3770 and 3790 batch data interchange LUs.

Default action: terminate the task abnormally.

INVMPSTZ

occurs if the specified map is too wide or too long for the terminal.

Default action: terminate the task abnormally.

INVPARTN

occurs if the specified partition is not defined in the partition set associated with the application program.

Default action: terminate the task abnormally.

INVREQ

occurs if a RECEIVE MAP command is issued in a nonterminal task; these tasks do not have a TIOA or a TCTTE.

Default action: terminate the task abnormally.

MAPFAIL

occurs if the data to be mapped has a length of zero or does not contain a set-buffer-address (SBA) sequence. It applies only to 3270 devices. The receiving data area contains the unmapped input data stream. The amount of unmapped data moved to the user's area is limited to the length specified in the LENGTH option. The input map is not set to nulls.

This condition also arises if a program issues a RECEIVE MAP command to which the terminal operator responds by pressing a CLEAR or PA key, or by pressing ENTER or a PF key without entering data.

Default action: terminate the task abnormally.

PARTNFAIL

occurs if the terminal operator attempts to enter data more than three times in a partition other than that specified by the INVPARTN option.

Default action: terminate the task abnormally.

RDATT

occurs if a RECEIVE MAP command is terminated by the operator using the ATTN key rather than the RETURN key. It applies only to the 2741 Communications Terminal, and only if 2741 read attention support has been generated for CICS.

Default action: ignore the condition.

UNEXPIN

occurs when unexpected or unrecognized data is received. This only applies to batch data interchange terminals.

Default action: terminate the task abnormally.

RECEIVE PARTN

Receive data from an 8775 terminal partition. This command is only available on standard and full BMS. For further information about BMS, see the *CICS Application Programming Guide*.

```
▶▶—RECEIVE PARTN(data-area)—◀◀
```

Standard and full BMS:

```
▶▶—INTO(data-area)—LENGTH(data-value)—◀◀
   |—SET(ptr-ref)—|
   |—ASIS—|
```

Conditions: EOC, EODS, INVPARTN, INVREQ, LENGERR

Description

RECEIVE PARTN reads data from a partition on an 8775 terminal. It indicates which partition the data came from, and puts the data into the INTO or the SET data area. You can then treat the data as though it had originated from a terminal in base (unpartitioned) state.

Following a RECEIVE PARTN command, the inbound cursor position is placed in EIBCPOSN, and the terminal attention identifier (AID) placed in EIBAID. EIBAID and EIBCPOSN are also updated at task initiation for non-ATI tasks as well as after each terminal control and BMS input.

See Appendix K, “BMS macro summary” on page 411 for map definitions.

If data is to be received, you must specify either the INTO or the SET option. If a RECEIVE is issued purely to detect an attention identifier (AID), you can omit both the INTO and the SET options.

Options

ASIS

specifies that lowercase characters in the 3270 input data stream are not translated to uppercase; this allows the current task to receive a message containing both uppercase and lowercase data.

The ASIS option has no effect on the first RECEIVE command of a transaction, or if the screen contains data prior to a transaction being initiated. For example, if a transaction is initiated by another transaction, and begins by receiving data originally output by that transaction, it cannot suppress uppercase translation on the data. This

data is read and translated in preparation for the next task and the first RECEIVE command in that task retrieves the translated data.

Note: If you are using a katakana terminal, you might see some messages containing mixed English and katakana characters. That is because katakana terminals cannot display mixed-case output. Uppercase characters in the data stream appear as uppercase English characters, but lowercase characters appear as katakana characters. If this happens, ask your system programmer to specify MSGCASE=UPPER in the system initialization parameters so that messages contain uppercase characters only.

INTO(*data-area*)

specifies the area into which the input data stripped of partition controls is to be written. The length of this area must be specified by the LENGTH option. If the area is not large enough to hold the input data, the input data is truncated, and the LENGERR condition raised. The length option data area is set to the length of data received, prior to any truncation.

LENGTH(*data-value*)

specifies the length of the data to be formatted as a halfword binary value. It must be set to the length of any INTO area prior to the command. After the command, BMS sets the LENGTH option to the length of data received prior to any truncation if the INTO area is too small.

For a description of a safe upper limit, see “LENGTH options” on page 5.

PARTN(*data-area*)

is set to the name (1–2 characters) of the input partition. The partition can be defined either by using RDO or by program autoinstall when the partition is first used.

Default action: truncate the data to fit within the INTO area.

SET(*ptr-ref*)

specifies the pointer that is to be set to the address of the 12-byte prefix to the mapped data. The pointer reference is valid until the next receive command or the end of task.

If DATALOCATION(ANY) is specified in the RDO PROGRAM resource definition for the application program, the address of the data may be above or below the 16MB line.

If DATALOCATION(BELOW) is specified in the RDO PROGRAM resource definition for the application program, and the data resides above the 16MB line, the data is copied below the 16MB line, and the address of this copy is returned.

If TASKDATAKEY(USER) is specified in the RDO TRANSACTION resource definition for the executing transaction, and storage protection is active, the data returned is in user-key DSA (UDSA or EUDSA); otherwise the data is returned in a CICS-key DSA (CDSA or ECDSA).

Conditions

Some of the following conditions can occur in combination. If more than one occurs, only the first one is passed to the application program.

EOC

occurs if the request/response unit (RU) is received with the end-of-chain (EOC) indicator set. It applies only to logical units.

Default action: ignore the condition.

EODS

occurs if no data is received (only an FMH). It applies only to 3770 batch LUs and to 3770 and 3790 batch data interchange LUs.

Default action: terminate the task abnormally.

INVPARTN

occurs if the specified partition is not defined in the partition set associated with the application program.

Default action: terminate the task abnormally.

INVREQ

occurs if a RECEIVE PARTN command is issued in a nonterminal task; these tasks do not have a TIOA or a TCTTE.

Default action: terminate the task abnormally.

LENGERR

occurs if the INTO area of a RECEIVE PARTN command is not large enough to hold the input data.

RELEASE

Release a loaded program, table, or mapset.

```
►►—RELEASE—PROGRAM(name)—►►
```

Conditions: INVREQ, NOTAUTH, PGMIDERR

Note for dynamic transaction routing

Using RELEASE of a program LOADED with HOLD could create inter-transaction affinities that adversely affect the use of dynamic transaction routing. See the *CICS Application Programming Guide* for more information about transaction affinities.

Description

RELEASE releases the program, table, or mapset previously loaded by a LOAD command. This means that the issuing task can no longer use the resource unless another LOAD is issued.

If the HOLD option is specified in the LOAD command, the loaded resource is not released at the end of the task. It can only be released by a RELEASE command. This RELEASE command may be issued by the task that loaded the resource or by any other task.

If the HOLD option is not specified in the LOAD command, the loaded resource is released at the end of the task. It may, however, be released before this by the task that loaded the resource issuing a RELEASE command.

Options

PROGRAM(*name*)

specifies the identifier (1–8 characters) of a program, table, or mapset to be released. The specified name must have been defined as a program to CICS.

Conditions

INVREQ

RESP2 values:

- 5 An invalid attempt is made by the program to release itself. A RELEASE command for the program that contains this command is allowed only when a corresponding LOAD command for the program has been issued from the same task, or when a LOAD command with the HOLD option has been issued from another task.

- 6 The command is issued for a program that is not loaded.
- 7 The command is issued for a program that was loaded, without the HOLD option, by another task.
- 17 The program is defined with RELOAD=YES. It must be released by a FREEMAIN rather than a RELEASE command.
- 30 The program manager domain has not yet been initialized. This is probably due to a release request having been made in a first stage PLT.

Default action: terminate the task abnormally.

NOTAUTH

occurs when a resource security check has failed on PROGRAM(*name*).

Default action: terminate the task abnormally.

PGMIDERR

RESP2 values:

- 1 A program, table, or mapset has no entry in the PPT.
- 2 A program, table, or mapset is disabled.
- 9 The installed program definition is for a remote program.

Default action: terminate the task abnormally.

Examples

The following example shows how to release an application program, called PROG4, loaded in response to a LOAD command:

```
EXEC CICS RELEASE PROGRAM('PROG4')
```

REPL (DL/I)

Replace one or more segments.

```

▶▶ REPL ┌──┴──┐ ┌──┴──┐
        │  USING PCB(integer-expr)  │
        └──┬──┘ └──┬──┘
  
```

For each parent segment:

```

▶▶ ┌──┴──┐ ┌──┴──┐ ┌──┴──┐ ┌──┴──┐ ┌──┴──┐ ┌──┴──┐ ┌──┴──┐ ┌──┴──┐
   │  VARIABLE  │ │ SEGMENT(name) │ │ SEGLENGTH(integer-expr) │ │ OFFSET(integer-expr) │
   └──┬──┘ └──┬──┘ └──┬──┘ └──┬──┘ └──┬──┘ └──┬──┘ └──┬──┘ └──┬──┘
   ┌──┴──┐ ┌──┴──┐
   │  FROM(data-area)  │
   └──┬──┘ └──┬──┘
  
```

For the object segment:

```

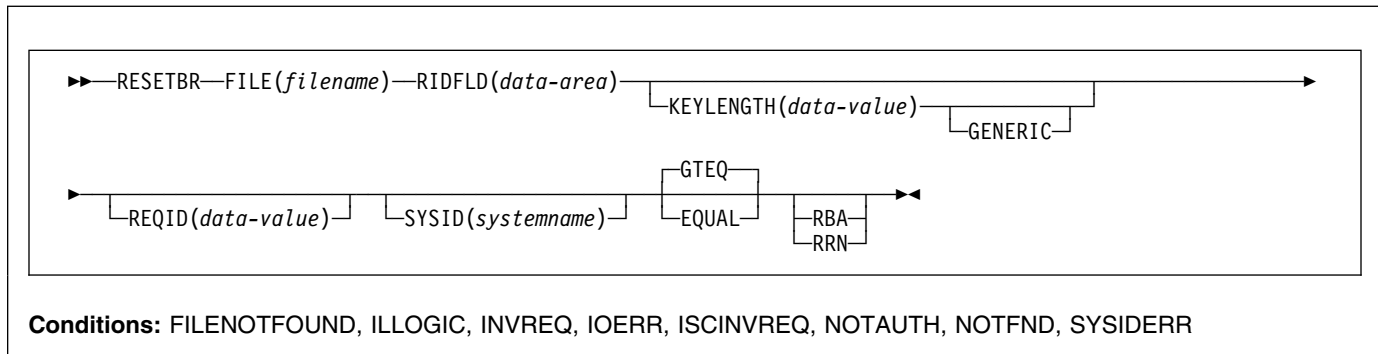
▶▶ ┌──┴──┐ ┌──┴──┐ ┌──┴──┐ ┌──┴──┐ ┌──┴──┐ ┌──┴──┐ ┌──┴──┐ ┌──┴──┐
   │  VARIABLE  │ │ SEGMENT(name) │ │ SEGLENGTH(integer-expr) │ │ OFFSET(integer-expr) │
   └──┬──┘ └──┬──┘ └──┬──┘ └──┬──┘ └──┬──┘ └──┬──┘ └──┬──┘ └──┬──┘
   ┌──┴──┐ ┌──┴──┐
   │  FROM(data-area)  │
   └──┬──┘ └──┬──┘
  
```

Description

The full form of the command is REPLACE. See *DL/I DOS/VS Application Programming: High Level Programming Interfaces* for details.

RESETBR

Reset start of browse.



Description

RESETBR specifies, during a browse, the record in a file or data table on a local or a remote system, where you want the browse to be repositioned.

When browsing a VSAM file or data table, you can use this command not only to reposition the browse (which can be achieved more simply by modifying the RIDFLD data area on a READNEXT or READPREV command), but also to change its characteristics from those specified on STARTBR, without ending the browse. The characteristics that may be changed are those specified by the GENERIC, GTEQ, and RBA options.

When browsing a DAM file, you can include this command at any time prior to issuing any other browse command. It is similar to an ENDBR–STARTBR sequence (but with less function), and gives the DAM user the sort of skip sequential capability that is available to VSAM users through use of the READNEXT command.

If a RESETBR request specifies the precise key at which the browse is to start (that is, it specifies a full key and the EQUAL keyword) the record returned on the following READNEXT (or READPREV) may not be the same as the record specified by the RESETBR for a file opened in VSAM NSR mode. This can occur because the initial record specified on the RESETBR command can be deleted by another transaction in between the RESETBR completing and a READNEXT or READPREV being issued. In VSAM LSR mode, the initial record cannot be deleted between the RESETBR and the READNEXT.

Options

EQUAL

specifies that the search is satisfied only by a record having the same key (complete or generic) as that specified in the RIDFLD option.

FILE(filename)

(VSAM and data table) specifies the name of the file to be accessed.

If SYSID is specified, the data set to which this file refers is assumed to be on a remote system irrespective of whether the name is defined in the FCT. Otherwise, the FCT entry is used to find out whether the data set is on a local or a remote system.

GENERIC

(VSAM KSDS, path or data table) specifies that the search key is a generic key whose length is specified in the KEYLENGTH option. The search for a record is satisfied when a record is found that has the same starting characters (generic key) as those specified.

GTEQ

(VSAM and data table) specifies that if the search for a record having the same key (complete or generic) as that specified in the RIDFLD option is unsuccessful, the first record having a greater key is retrieved. Use this option only with keyed or RRN.

KEYLENGTH(data-value)

specifies the length (halfword binary) of the key that has been specified in the RIDFLD option, except when RBA or RRN is specified, in which case KEYLENGTH is not valid.

This option must be specified if GENERIC is specified, and it can be specified whenever a key is specified. If the length specified is different from the length defined for the data set and the operation is not generic, the INVREQ condition occurs.

The INVREQ condition also occurs if a RESETBR command specifies GENERIC, and the KEYLENGTH is not less than that specified in the VSAM definition.

If KEYLENGTH(0) is used with the object of reading the first record in the data set, the GTEQ option must also be specified. If EQUAL is specified either explicitly or by default with KEYLENGTH(0), the results of the STARTBR are unpredictable.

RBA

(VSAM KSDS or ESDS base data sets, or CICS-maintained data tables only, not paths) specifies that the record identification field specified in the RIDFLD option contains a relative byte address. Use this option only when browsing a KSDS and using relative byte addresses instead of keys to identify the records.

You cannot use RBA for:

- User-maintained data tables
- KSDS files that hold more than 4GB

REQID(*data-value*)

specifies as a halfword binary value a unique request identifier for a browse, used to control multiple browse operations on a data set. If this option is not specified, a default value of zero is assumed.

RIDFLD(*data-area*)

specifies the record identification field. The contents can be a key, a relative byte address, or a relative record number (for VSAM data sets), or a block reference, physical key, and a deblocking argument (for DAM data sets). For a relative byte address or a relative record number, the format of this field must be fullword binary. For a relative byte address, the RIDFLD can be greater than or equal to zero. For a relative record number, the RIDFLD can be greater than or equal to 1.

For VSAM, a full record id of X'FF's indicates that the browse is to be positioned at the end of the data set in preparation for a backwards browse using READPREV commands.

RRN

(VSAM RRDS) specifies that the record identification field specified in the RIDFLD option contains a relative record number.

SYSID(*systemname*)

specifies the name of the system to which the request is directed.

If you specify SYSID, and omit both RBA and RRN, you must also specify KEYLENGTH; it cannot be found in the FCT.

Conditions

Note: RESP2 values are not set for files that are on remote systems.

FILENOTFOUND

RESP2 values:

- 1** A file name referred to in the FILE option cannot be found in the FCT.

Default action: terminate the task abnormally.

ILLOGIC

RESP2 values VSAM):

- 110** A VSAM error occurs that does not fall within one of the other CICS response categories.

(See EIBRCODE in the EXEC interface block, starting on 373.)

Default action: terminate the task abnormally.

INVREQ

RESP2 values:

- 25** The KEYLENGTH and GENERIC options are specified, and the length specified in the KEYLENGTH option is greater than or equal to the length of a full key.

- 26** The KEYLENGTH option is specified (but the GENERIC option is not specified), and the specified length does not equal the length defined for the data set to which this file refers.

- 36** The REQID, if any, does not match that of any successful STARTBR command.

- 42** The KEYLENGTH and GENERIC options are specified, and the length specified in the KEYLENGTH option is less than zero.

- # 49** An attempt was made to browse a file that is an invalid file type.

Default action: terminate the task abnormally.

IOERR

RESP2 values:

- 120** There is an I/O error during the file control operation. An I/O error is any unusual event that is not covered by a CICS condition.

For VSAM files, IOERR normally indicates a hardware error.

(Further information is available in the EXEC interface block, defined on 373.)

Default action: terminate the task abnormally.

ISCINVREQ

RESP2 values:

- 70** The remote system indicates a failure that does not correspond to a known condition.

RESETBR

Default action: terminate the task abnormally.

NOTAUTH

RESP2 values:

- 101** A resource security check has failed on FILE(filename).

Default action: terminate the task abnormally.

NOTFND

RESP2 values:

- 80** An attempt to retrieve a record based on the search argument provided is unsuccessful.

NOTFND can also occur if a generic RESETBR with KEYLENGTH(0) specifies the EQUAL option.

Default action: terminate the task abnormally.

SYSIDERR

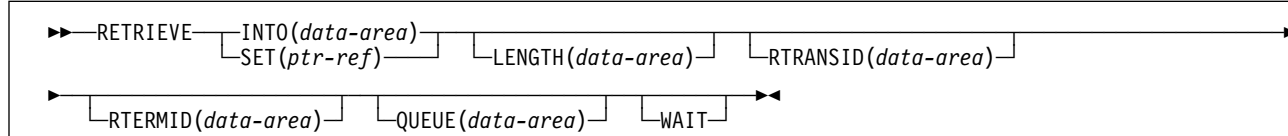
RESP2 values:

- 130** The SYSID option specifies a name that is neither the local system nor a remote system (made known to CICS by defining an RDO CONNECTION). SYSIDERR also occurs when the link to the remote system is closed

Default action: terminate the task abnormally.

RETRIEVE

Retrieve data stored for a task.



Conditions: ENDDATA, ENVDEFERR, INVREQ, IOERR, LENGERR, NOTFND

Note for dynamic transaction routing

Using RETRIEVE with WAIT could create inter-transaction affinities that adversely affect the use of dynamic transaction routing. See the *CICS Application Programming Guide* for more information about transaction affinities.

Description

The RETRIEVE command retrieves data stored by expired START commands. It is the only method available for accessing such data.

A task that is not associated with a terminal can access only the single data record associated with the original START command; it does so by issuing a RETRIEVE command. The storage occupied by the data associated with the task is released on execution of the RETRIEVE command, or on termination of the task if no RETRIEVE command is executed prior to termination.

A task that is associated with a terminal can access all data records associated with all expired START commands having the same transaction identifier and terminal identifier as this task, that is the task issuing the RETRIEVE command; it does so by issuing consecutive RETRIEVE commands. Expired data records are presented to the task on request in expiration-time sequence, starting with any data stored by the command that started the task, and including data from any commands that have expired since the task started. Each data record is retrieved from temporary storage using the REQID of the original START command as the identification of the record in temporary storage.

When all expired data records have been retrieved, the ENDDATA condition occurs. The storage occupied by the single data record associated with a START command is released after the data has been retrieved by a RETRIEVE command; any storage occupied by data that has not been retrieved is released when the CICS system is terminated.

If the retrieved data contains FMHs (Function Management Headers), as specified by the FMH option on the associated START command, field EIBFMH in the EIB is set to X'FF'. If no FMH is present, EIBFMH is set to X'00'.

Options

INTO(data-area)

specifies the user data area into which retrieved data is to be written.

LENGTH(data-area)

specifies a halfword binary value to define the length of the data area the retrieved data is written into.

If you specify the INTO option, the argument must be a data area that specifies the maximum length of data that the program is prepared to handle. If the value specified is less than zero, zero is assumed. If the length of the data exceeds the value specified, the data is truncated to that value and the LENGERR condition occurs. On completion of the retrieval operation, the data area is set to the original length of the data.

If you specify the SET option, the argument must be a data area. On completion of the retrieval operation, the data area is set to the length of the data.

For a description of a safe upper limit, see “LENGTH options” on page 5.

QUEUE(data-area)

specifies the 8-character area for the temporary storage queue name that may be accessed by the transaction issuing the RETRIEVE command.

RTERMID(data-area)

specifies a 4-character area that can be used in the TERMID option of a START command that may be executed subsequently.

RETRIEVE

RTRANSID(*data-area*)

specifies a 4-character area that can be used in the TRANSID option of a START command that may be executed subsequently.

SET(*ptr-ref*)

specifies the pointer reference to be set to the address of the retrieved data.

If DATALOCATION(ANY) is specified in the RDO PROGRAM resource definition for the application program, the address of the data may be above or below the 16MB line.

If DATALOCATION(BELOW) is specified in the RDO PROGRAM resource definition for the application program, and the data resides above the 16MB line, the data is copied below the 16MB line, and the address of this copy is returned.

If TASKDATAKEY(USER) is specified in the RDO TRANSACTION resource definition for the executing transaction, and storage protection is active, the data returned is in user-key DSA (JDSA or EDSA); if TASKDATAKEY(CICS) is specified and storage protection is active, the data returned is in CICS-key DSA (CDSA or ECDSA).

If you use SET you must also include LENGTH.

WAIT

specifies that, if all expired data records have already been retrieved, the task is to be put into a wait state until further expired data records become available. Although this means that the ENDDATA condition is not raised at the time the RETRIEVE command is issued, it is raised later if CICS enters shutdown or if the task is subject to deadlock time-out and it waits for longer than the deadlock time-out interval. (See the DTIMOUT option of RDO TRANSACTION resource definition in the *CICS Resource Definition Guide* for more information.)

An attempt to issue RETRIEVE WAIT during shutdown leads to an AICB abend if there is no data record already available to satisfy the request.

If you use WAIT, you must have at least one other option.

Conditions

ENDDATA

occurs in any of the following situations:

- No more data is stored for the task issuing a RETRIEVE command. It can be considered a normal end-of-file response when retrieving data records sequentially.
- The RETRIEVE command is issued by a task that is started by a START command that did not specify any of the data options FROM, RTRANSID, RTERMID, or QUEUE.

- The RETRIEVE command is issued by a nonterminal task that was not created as a result of a START command.
- WAIT was specified and the task was waiting for a data record but none became available before the deadlock time-out interval (see the DTIMOUT option of RDO DEFINE TRANSACTION).
- WAIT was specified and the task was waiting when CICS entered shutdown. An attempt to issue RETRIEVE WAIT during shutdown leads to an AICB abend if there is no data record already available to satisfy the request.
- A RETRIEVE command with the WAIT option is issued when no data is available; the task was initiated by a START command that specified an APPC connection or terminal in the RTERMID option.

Default action: terminate the task abnormally.

ENVDEFERR

occurs when a RETRIEVE command specifies an option not specified by the corresponding START command.

Default action: terminate the task abnormally.

INVREQ

occurs if the RETRIEVE command is not valid for processing by CICS.

Default action: terminate the task abnormally.

IOERR

occurs if an input/output error occurs during a RETRIEVE operation. The operation can be retried by reissuing the RETRIEVE command.

Default action: terminate the task abnormally.

LENGERR

occurs if the length specified is less than the actual length of the stored data.

Default action: terminate the task abnormally.

NOTFND

occurs in any of the following situations:

- A RETRIEVE command is issued, but some prior task retrieved the data stored under the request identifier directly through temporary storage requests and then released the data.
- The request identifier associated with the START command is not unique, so when a RETRIEVE command is issued, CICS cannot find the data.

Default action: terminate the task abnormally.

Examples

The following example shows how to retrieve data stored by a START command for the task, and store it in the user-supplied data area called DATAFLD.

```
EXEC CICS RETRIEVE  
  INTO(DATAFLD)  
  LENGTH(LENG)
```

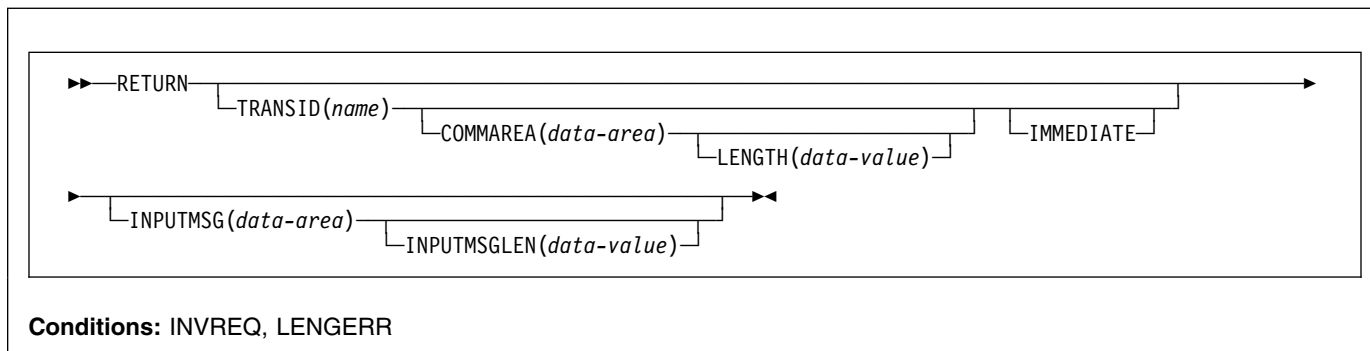
The following example shows how to request retrieval of a data record stored for a task into a data area provided by CICS; the pointer reference (PREF) specified by the SET option is set to the address of the storage area reserved for the data record.

```
EXEC CICS RETRIEVE  
  SET(PREF)  
  LENGTH(LENG)
```

RETURN

RETURN

Return program control.



Description

RETURN returns control from an application program either to an application program at the next higher logical level, or to CICS.

The LENGTH option specifies the length of the data to be passed. The LENGTH value being passed must not be greater than the length of the data area specified in the COMMAREA option. If it is, the results are unpredictable and may result in a LENGERR condition, as described in the section about passing data to other programs in the *CICS Application Programming Guide*.

The valid range for the COMMAREA length is 0 through 32 763 bytes. If the length provided is outside this range, the LENGERR condition occurs. The COMMAREA and IMMEDIATE options can be used only when the RETURN command is returning control to CICS; otherwise, the INVREQ condition occurs.

No resource security checking occurs on the RETURN TRANSID command. However, transaction security checking is still available when CICS attaches the returned transaction.

Options

COMMAREA(*data-area*)

specifies a communication area that is to be made available to the next program that receives control. In a COBOL receiving program, you must give this data area the name DFHCOMMAREA. (See the *CICS Application Programming Guide* for more information about the CICS COMMAREA.) Because the data area is freed before the next program starts, a copy of the data area is created and a pointer to the copy is passed.

The communication area specified is passed to the next program that runs at the terminal. To ensure that the

communication area is passed to the correct program, include the IMMEDIATE option.

This option is valid only on a RETURN command issued by a program at the highest logical level, that is, a program returning control to CICS.

IMMEDIATE

ensures that the transaction specified in the TRANSID option is attached as the next transaction regardless of any other transactions enqueued by ATI for this terminal. The next transaction starts immediately and appears to the operator as having been started by terminal data. If the terminal is using bracket protocol, the terminal is also held in bracket. This option is valid only on a RETURN command issued by a program at the highest logical level, that is a program returning control to CICS.

Note that in a multi region environment, using IMMEDIATE does not affect the transaction definition as this is still found in the terminal-owning region (TOR).

INPUTMSG(*data-area*)

specifies data to be passed either to another transaction, identified by the TRANSID option, or to a calling program in a multiprogram transaction. You can also use INPUTMSG when returning control to CICS from a user-written dynamic transaction routing program, when you might want to modify the initial input.

In all cases, the data in the INPUTMSG data area is passed to the first program to issue a RECEIVE command following the RETURN.

See the *CICS Application Programming Guide* for more information and illustrations about the use of INPUTMSG.

INPUTMSGLEN(*data-value*)

specifies a halfword binary value to be used with INPUTMSG.

LENGTH(*data-value*)

specifies a halfword binary value that is the length in bytes of the COMMAREA. For a description of a safe upper limit, see “LENGTH options” on page 5.

TRANSID(*name*)

specifies the transaction identifier (1–4 characters) to be used with the next input message entered from the terminal with which the task that issued the RETURN command has been associated. The specified name must have been defined as a transaction to CICS.

If TRANSID is specified for a program running on a terminal that is defined with a permanent transaction ID, the terminal's permanent transaction is initiated next rather than the transaction specified on the RETURN.

If you specify a TRANSID of binary zeros, the transaction identifier for the next program to be associated with the terminal may be determined from subsequent input from the terminal. Issuing a RETURN with a TRANSID of binary zeros and a COMMAREA can cause unpredictable results if the next transaction is not coded to handle the COMMAREA or if it receives a COMMAREA not intended for it.

If you specify TRANSID on a program that is not at the highest level, and there is a subsequent error on COMMAREA or INPUTMSG on the final RETURN, the TRANSID is cleared.

The next transaction identifier is also cleared on an abnormal termination of the transaction.

If IMMEDIATE is specified with this option, control is passed to the transaction specified in the TRANSID option in preference to any transactions enqueued by ATI.

If IMMEDIATE is not specified with this option, an ATI initiated transaction of the same name enqueued to the terminal nullifies this option.

This option is not valid if the transaction issuing the RETURN command is not associated with a terminal, or is associated with an APPC logical unit.

8 A RETURN command with the INPUTMSG option is issued for a program that is not associated with a terminal, or that is associated with an APPC logical unit, or an IRC session.

30 PG domain not initialized. Parameters are not allowed on the EXEC RETURN statement in first stage PLT programs.

200 A RETURN command is issued with an INPUTMSG option by a program invoked by DPL.

Default action: terminate the task abnormally.

LENGERR

RESP2 values:

11 The COMMAREA length is less than 0 or greater than 32763.

26 The COMMAREA ADDRESS passed was zero, but the commarea length was non-zero.

27 The INPUTMSG LENGTH was less than 0 or greater than 32767.

Default action: terminate the task abnormally.

Conditions**INVREQ**

RESP2 values:

1 A RETURN command with the TRANSID option is issued in a program that is not associated with a terminal.

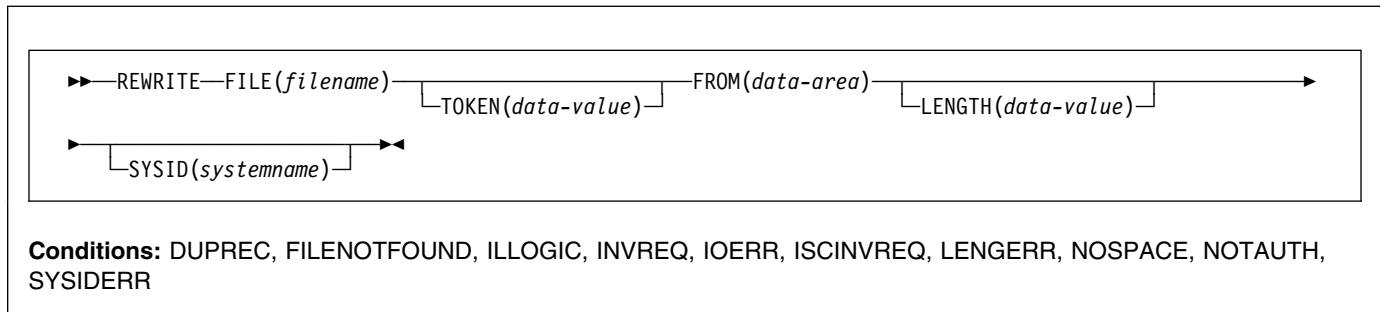
2 A RETURN command with the COMMAREA or IMMEDIATE option is issued by a program that is not at the highest logical level.

4 A RETURN command with the TRANSID option is issued in a program that is associated with an APPC logical unit.

REWRITE

REWRITE

Update a record in a file.



Conditions: DUPREC, FILENOTFOUND, ILLOGIC, INVREQ, IOERR, ISCVREQ, LENGERR, NOSPACE, NOTAUTH, SYSIDERR

Description

REWRITE updates a record in a file on a local or a remote system. You must always precede this command with a read with the UPDATE option.

For VSAM data sets, you must not change the key field in the record.

When this command is used to update a record in a CICS-maintained data table, the update is made to both the source VSAM KSDS and the in-memory data table. The details of the command for a CICS-maintained table are the same as for a VSAM KSDS.

When this command is used to update a record in a user-maintained data table, the update is made to the in-memory data table.

Options

FILE(filename)

specifies the of the file to be accessed.

If SYSID is specified, the data set to which this file refers is assumed to be on a remote system irrespective of whether the name is defined in the FCT. Otherwise, the FCT entry is used to find out whether the data set is on a local or a remote system.

FROM(data-area)

specifies the record that is to be written to the data set referred to by this file.

LENGTH(data-value)

specifies the actual length, as a halfword binary value, of the record to be written.

This option must be specified if SYSID is specified.

This option must also be specified for a file defined as containing variable-length records; which includes user-maintained data tables. It need not be specified if the file contains fixed-length records. You are

recommended to include it, however, because it causes a check to be made to ensure that the length of data being written is equal to that defined for the data set. If the lengths are not equal, the LENGERR condition occurs.

SYSID(systemname)

specifies the name of the system to which the request is directed.

TOKEN(data-area)

specifies as a fullword binary value a unique request identifier for a REWRITE, used to associate it with a previous READ, READNEXT, or READPREV command that specified UPDATE.

Conditions

Note: RESP2 values are not set for files that are on remote systems.

DUPREC

RESP2 values:

- 150** An attempt is made to rewrite a record to a data set whose upgrade set has an alternate index with the UNIQUEKEY attribute, if the corresponding alternate key already exists in the alternate index.

Default action: terminate the task abnormally.

FILENOTFOUND

RESP2 values:

- 1** A file name referred to in the FILE option cannot be found in the FCT.

Default action: terminate the task abnormally.

ILLOGIC

RESP2 values: (VSAM)

110 A VSAM error occurs that does not fall within one of the other CICS response categories.

(See EIBRCODE in the EXEC interface block; refer to Appendix A, "EXEC interface block" on page 373 for details.)

Default action: terminate the task abnormally.

INVREQ

RESP2 values:

30 A REWRITE command is issued without a token and no previous READ for UPDATE (also without a token) can be found.

46 A REWRITE command has attempted to change the length of a DAM variable length record or block.

47 A REWRITE instruction includes a token whose value cannot be matched against any token in use for an existing read for UPDATE request.

48 An attempt is made to function-ship a request that includes a TOKEN keyword.

49 An attempt was made to rewrite to a file that is an
invalid file type.

Default action: terminate the task abnormally.

IOERR

RESP2 values:

120 An I/O error occurred during the file control operation. An I/O error is any unusual event that is not covered by a CICS condition.

For VSAM files, IOERR normally indicates a hardware error.

(Further information is available in the EXEC interface block; refer to Appendix A, "EXEC interface block" on page 373 for details.)

Default action: terminate the task abnormally.

ISCVREQ

RESP2 values:

70 The remote system indicates a failure that does not correspond to a known condition.

Default action: terminate the task abnormally.

LENGERR

RESP2 values:

10 The LENGTH option is not specified for a file with variable-length records, or for a DAM file with undefined format records.

12 The length specified exceeds the maximum record size (of the source data set for a data-table); the record is truncated.

14 An incorrect length is specified for a file with fixed-length records.

Default action: terminate the task abnormally.

NOTFND

RESP2 values:

80 For user-maintained data tables, this condition occurs when an attempt to REWRITE a record has failed because the REWRITE is associated with a READ UPDATE request for a record that this transaction has deleted (using DELETE with RIDFLD) after it was read for update .

This may be caused by a logic error in the application program.

Default action: terminate the task abnormally.

NOSPACE

RESP2 values:

100 No space is available on the direct access device for adding the updated record to the data set.

103 For user-maintained data tables, this condition occurs if CICS is unable to get sufficient storage in the CICS address space to store the updated data table entry.

Default action: terminate the task abnormally.

NOTAUTH

RESP2 values:

101 A resource security check has failed on FILE(filename).

Default action: terminate the task abnormally.

SYSIDERR

RESP2 values:

130 The SYSID option specifies a name that is neither the local system nor a remote system (made known to CICS by defining an RDO CONNECTION). SYSIDERR also occurs when the link to the remote system is closed.

Default action: terminate the task abnormally.

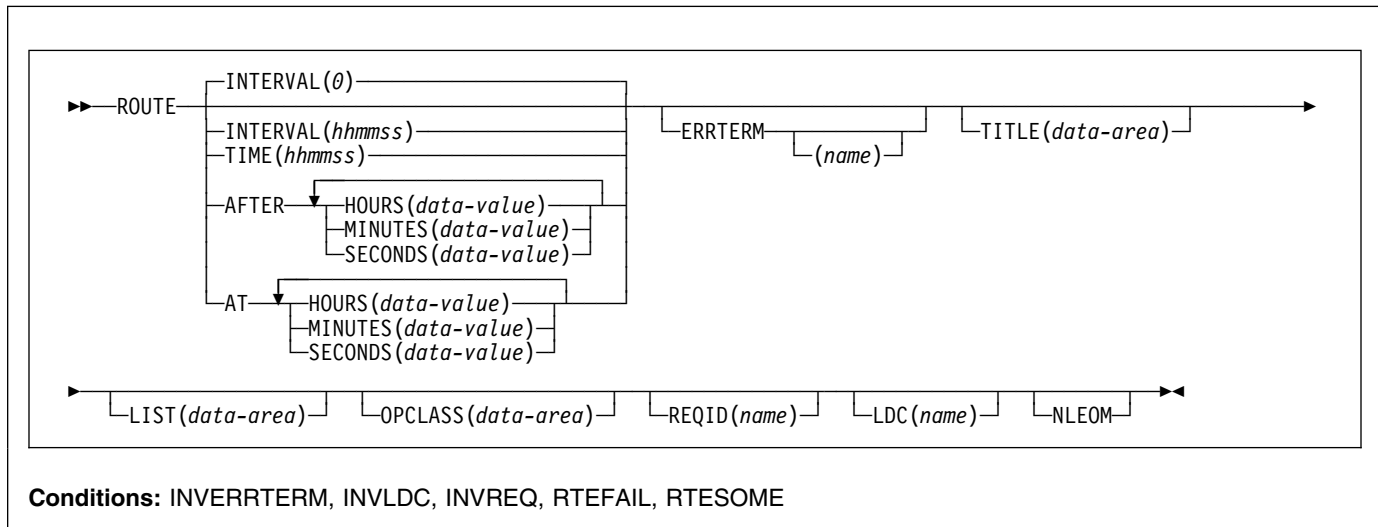
Examples

For example:

```
EXEC CICS REWRITE
      FROM(RECORD)
      FILE('MASTER')
      TOKEN(APTOK)
```

ROUTE

Route a BMS message. (This command is only available on full BMS. The *CICS Application Programming Guide* has further information about BMS.)



Description

ROUTE routes a BMS logical message to one or more terminals or terminal operators.

The default is INTERVAL(0), but for C the default is AFTER HOURS(0) MINUTES(0) SECONDS(0).

Options

AFTER

specifies the amount of time to elapse before the route.

There are two ways to enter the time under AFTER and AT.

1. A combination of at least two of HOURS(0–99), MINUTES(0–59), and SECONDS(0–59). HOURS(1) SECONDS(3) would mean one hour and three seconds (the minutes default to zero).
2. As one of HOURS(0–99), MINUTES(0–5999), or SECONDS(0–359999). HOURS(1) means one hour. MINUTES(62) means one hour and two minutes. SECONDS(3723) means one hour, two minutes, and three seconds.

AT specifies the time of the route. For the ways to enter the time, see the AFTER option.

ERRTERM(name)

specifies the name of the terminal to be notified if the message is deleted because it is undeliverable. The message number, title identification, and destination are

indicated. If no name is specified, the originating terminal is assumed.

This option is effective only if PRGDLAY has been specified in the system initialization parameters. See the *CICS System Definition Guide* for more information about the PRGDLAY system initialization parameter.

HOURS(data-value)

specifies a fullword binary value in the range 0–99. This is a suboption of the AFTER and AT options. For its use and meaning, see the AFTER option.

INTERVAL(hhmmss)

specifies the interval of time after which the data is to be transmitted to the terminals specified in the ROUTE command. The **mm** and **ss** are in the range 0–59.

When using the C language, you are recommended to use the AFTER/AT HOURS, MINUTES, and SECONDS options as C does not provide a packed decimal data type. You may use INTERVAL, but if the value specified is **not** an integer constant, the application is responsible for ensuring that the value passed to CICS is in packed decimal format.

LDC(name) — logical units only

specifies a 2-character mnemonic to be used to determine the logical device code (LDC) to be transmitted in the FMH to the logical unit. The mnemonic identifies an LDC entry defined by a DFHTCT TYPE=LDC macro. See the *CICS Resource Definition Guide* for more information about the DFHTCT macro.

When an LDC is specified, BMS uses the device type, the page size, and the page status associated with the

LDC mnemonic to format the message. These values are taken from the extended local LDC table for the LU, if it has one. If the LU has only a local (unextended) LDC table, the values are taken from the system LDC table. The numeric value of the LDC is obtained from the local LDC table, unless this is an unextended table and the value is not specified, in which case it is taken from the system table.

If the LDC option is omitted, the LDC mnemonic specified in DFHMSD is used; see “DFHMSD” on page 427 for further details. If the LDC option has also been omitted from DFHMSD, the action depends on the type of logical unit, as follows:

3601 LU

The first entry in the local or extended local LDC table is used, if there is one. If a default cannot be obtained in this way, a null LDC numeric value (X'00') is used. The page size used is the value that is specified in the RDO TYPETERM options PAGESIZE and ALTPAGE, or (1,40) if such a value is not specified.

LUTYPE4 LU, batch LU, or batch data interchange LU

The local LDC table is not used to supply a default LDC; instead, the message is directed to the LU console. (Here, LU console means any medium on which the LU elects to receive such messages. For a batch data interchange LU, this does not imply sending an LDC in an FMH). The page size is obtained in the manner described for the 3601 LU.

For message routing, the LDC option of the ROUTE command takes precedence over all other sources. If this option is omitted and a route list is specified (LIST option), the LDC mnemonic in the route list is used; if the route list contains no LDC mnemonic, or no route list is specified, a default LDC is chosen as described above.

LIST(*data-area*)

specifies the data area that contains a list of terminals and operators to which data is to be directed. If this option is omitted, all terminals supported by BMS receive the data (unless the OPCLASS option is in effect). See the *CICS Application Programming Guide* for the format of the route list.

MINUTES(*data-value*)

specifies a fullword binary value in the range 0–59, when HOURS or SECONDS are also specified, or 0–5999 when MINUTES is the only option specified. This is a suboption of the AFTER and AT options. For its use and meaning, see the AFTER option.

NLEOM

specifies that data for a 3270 printer or a 3275 display with the printer adapter feature should be built with

blanks and new-line (NL) characters, and that an end-of-message (EM) character should be placed at the end of the data. As the data is printed, each NL character causes printing to continue on the next line, and the EM character terminates printing.

The option is ignored if the device receiving the message (direct or routed) is not one of those mentioned above.

If this option is used, buffer updating and attribute modification of fields previously written into the buffer are not allowed. CICS includes the ERASE option with every write to the terminal.

The NL character occupies a buffer position. A number of buffer positions, equivalent to the value of the RDO TYPETERM options PAGESIZE or ALTPAGE for that terminal, are unavailable for data. This may cause data to wrap around in the buffer; if this occurs, the PAGESIZE or ALTPAGE value must be reduced.

OPCLASS(*data-area*)

specifies the data area that contains a list of operator classes to which the data is to be routed. The classes are supplied in a 3-byte field, each bit position corresponding to one of the codes in the range 1 through 24 but in reverse order, that is, the first byte corresponds to codes 24 through 17, the second byte to codes 16 through 9, and the third byte to codes 8 through 1.

REQID(*name*)

specifies a prefix (2–character field) to be used as part of a temporary storage identifier for CICS message recovery. Only one prefix can be specified for each logical message. The default prefix is **.

BMS message recovery is provided for a logical message only if the PAGING option is specified in the BMS SEND commands, and if the syncpoint has been reached.

SECONDS(*data-value*)

specifies a fullword binary value in the range 0–59, when HOURS or MINUTES are also specified, or 0–359 999 when SECONDS is the only option specified. This is a suboption of the AFTER and AT options. For its use and meaning, see the AFTER option.

TIME(*hhmmss*)

specifies the time of day at which data is to be transmitted to the terminals specified in the ROUTE command.

When using the C language, you are recommended to use the AFTER/AT HOURS, MINUTES, and SECONDS options as C does not provide a packed decimal data type. You may use TIME, but if the value specified is **not** an integer constant, the application is responsible for ensuring that the value passed to CICS is in packed decimal format.

ROUTE

TITLE(*data-area*)

specifies the data area that contains the title to be used with a routing logical message. This title appears as part of the response to a page query command. See the *CICS Application Programming Guide* for the format of the title option.

Conditions

INVERRTERM

occurs if the terminal identifier specified in the ERRTERM option is not valid or is assigned to a type of terminal not supported by BMS.

Default action: terminate the task abnormally.

INVLDC

occurs if the specified LDC mnemonic is not included in the LDC list for the logical unit.

Default action: terminate the task abnormally.

INVREQ

RESP2 values:

- 4** Hours out of range
- 5** Minutes out of range
- 6** Seconds out of range
- 200** BMS commands are not supported for distributed program link.

also occurs (RESP2 not set) in the following situations:

- Bytes 10 through 15 of a route list entry do not contain blanks.

Default action: terminate the task abnormally.

RTEFAIL

occurs in any of the following situations:

- A ROUTE command would result in the message being sent only to the terminal that initiated the transaction.
- A ROUTE command is issued against a remote, shippable terminal that is not yet installed in the application-owning region.

Default action: return control to the application program at the point immediately following the ROUTE command.

RTESOME

occurs if any of the terminals specified by the ROUTE command options do not receive the message.

Default action: return control to the application program at the point immediately following the ROUTE command.

SCHD (DL/I)

Schedule the PSB.

```
▶▶SCHD—PSB(name)—▶▶
```

Description

The full form of the command is SCHEDULE. See *DL/I DOS/VS Application Programming: High Level Programming Interfaces* for details.

SEND (VTAM)

SEND (VTAM default)

Write data to a standard CICS supported terminal.

```
▶▶ SEND FROM(data-area) LENGTH(data-value)  
                                FLENGTH(data-value) WAIT ▶▶
```

Conditions: INVREQ, LENGERR, NOTALLOC

Description

SEND writes data to a terminal. This form of the send command can be used by all CICS-supported terminals for which the other SEND descriptions are not appropriate.

SEND (APPC)

Send data on an APPC mapped conversation.

```
▶▶ SEND CONVID(name) FROM(data-area) LENGTH(data-value)  
                                FLENGTH(data-value) INVITE CONFIRM  
                                LAST WAIT ▶▶  
▶ STATE(cvda) ▶▶
```

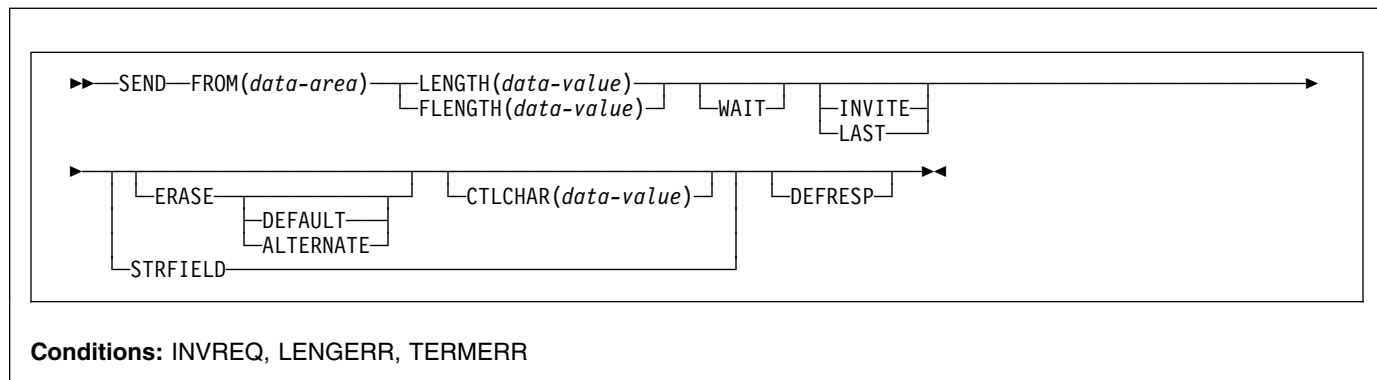
Conditions: INVREQ, LENGERR, NOTALLOC, SIGNAL, TERMERR

Description

SEND sends data to a conversation partner on an APPC mapped conversation.

SEND (LUTYPE2/LUTYPE3)

Write data to a 3270-display logical unit (LUTYPE2) or a 3270-printer logical unit (LUTYPE3).

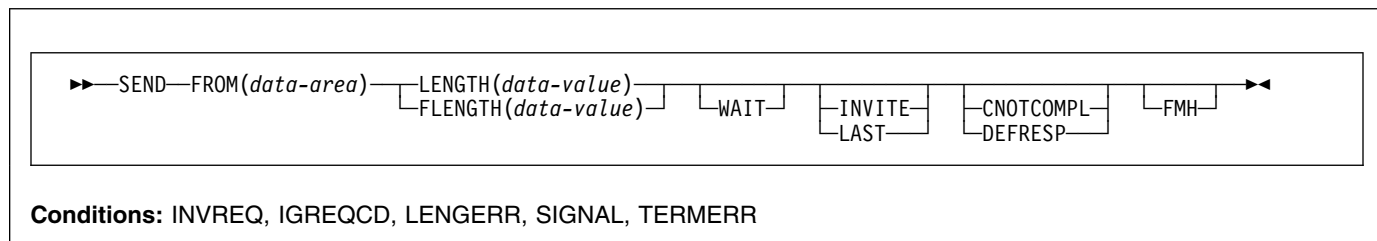


Description

SEND writes data to a terminal.

SEND (LUTYPE4)

Write data to a LUTYPE4 logical unit.



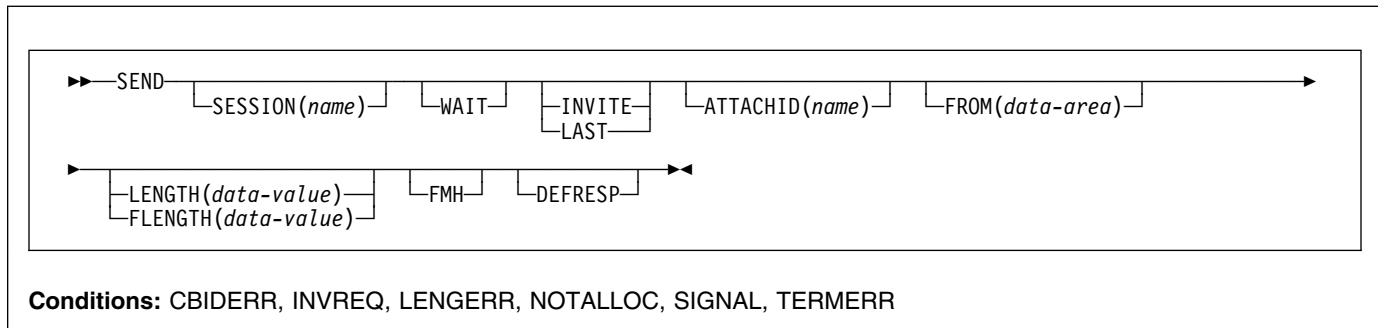
Description

SEND writes data to a terminal.

SEND (VTAM)

SEND (LUTYPE6.1)

Send data on an LUTYPE6.1 conversation.

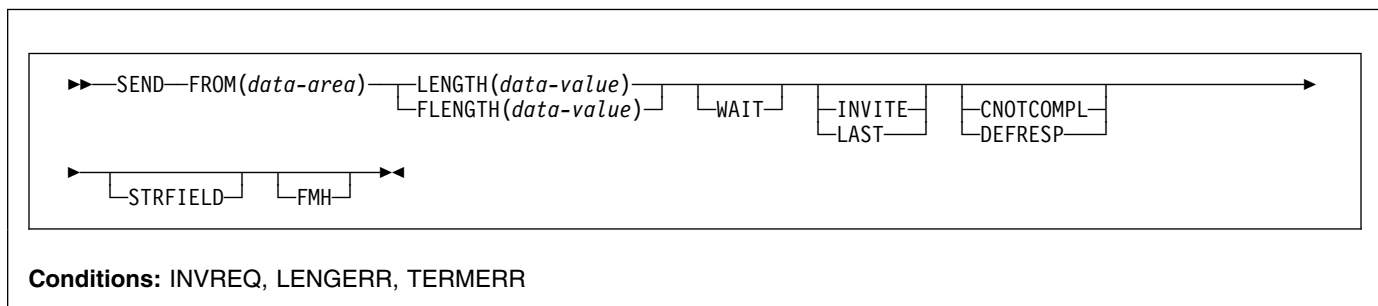


Description

SEND sends data to a conversation partner on an LUTYPE6.1 conversation.

SEND (SCS)

Write data to a 3270 SCS printer logical unit.

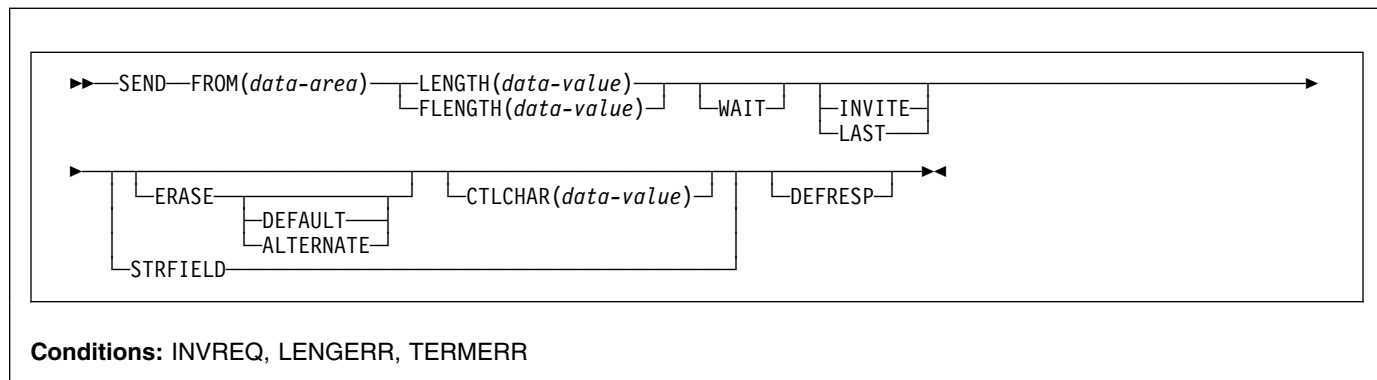


Description

SEND writes data to a logical unit. The SCS printer logical unit accepts a character string as defined by Systems Network Architecture (SNA).

SEND (3270 logical)

Write data to a 3270 logical unit.

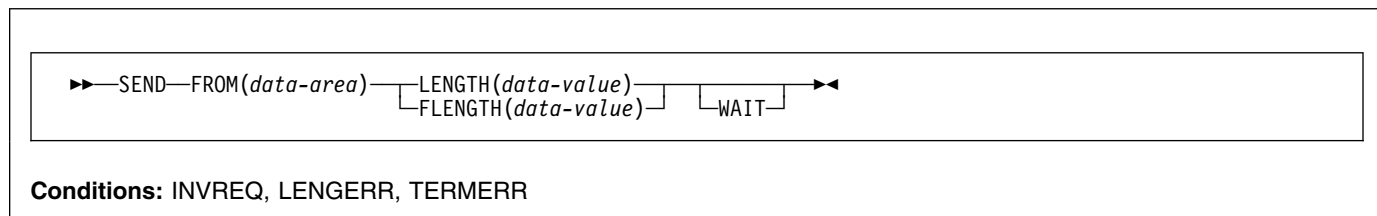


Description

SEND writes data to a terminal.

SEND (3600 pipeline)

Write data to a 3600 pipeline logical unit.



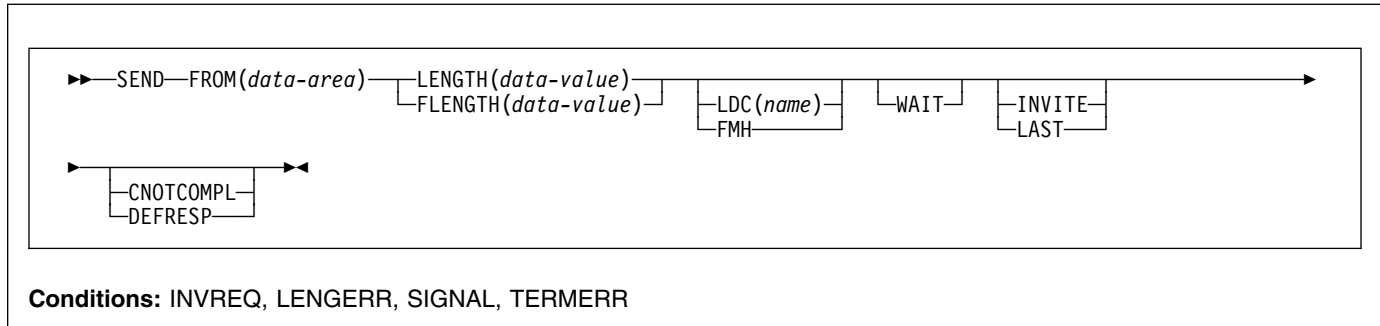
Description

SEND writes data to a terminal.

SEND (VTAM)

SEND (3600-3601)

Write data to a 3600 (3601) logical unit.



Description

SEND writes data to a terminal. This form of SEND also applies to the 3630 plant communication system.

A logical device code (LDC) is a code that can be included in an outbound FMH to specify the disposition of the data (for example, to which subsystem terminal it should be sent). Each code can be represented by a unique LDC mnemonic.

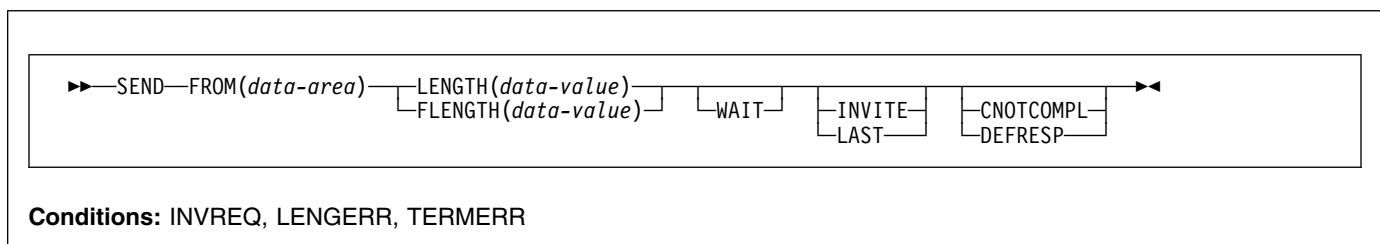
The installation can specify up to 256 2-character mnemonics for each TCTTE, and two or more TCTTEs can share a list of these mnemonics. Corresponding to each LDC mnemonic for each TCTTE is a numeric value (0 through 255).

A 3600 device and a logical page size are also associated with an LDC. "LDC" or "LDC value" is used in this book in reference to the code specified by the user. "LDC mnemonic" refers to the 2-character symbol that represents the LDC numeric value.

When the LDC option is specified, the numeric value associated with the mnemonic for the particular TCTTE, is inserted in the FMH. The numeric value associated with the LDC mnemonic is chosen by the installation, and is interpreted by the 3601 application program.

SEND (3600-3614)

Write data to a 3600 (3614) logical unit.



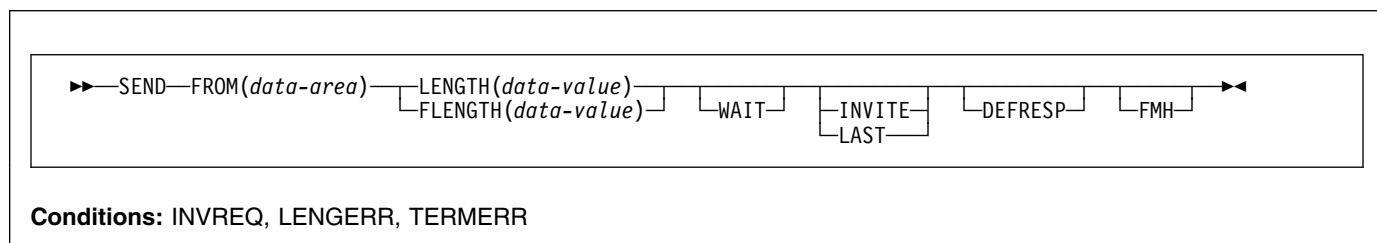
Description

SEND writes data to a terminal. The data stream and communication format used between a CICS application program and a 3614 is determined by the 3614. The application program is therefore device dependent when handling 3614 communication.

For further information about designing 3614 application programs for CICS, refer to the *CICS/DOS/VS IBM 4700/3600/3630 Guide*.

SEND (3650 interpreter)

Write data to a 3650 interpreter logical unit.

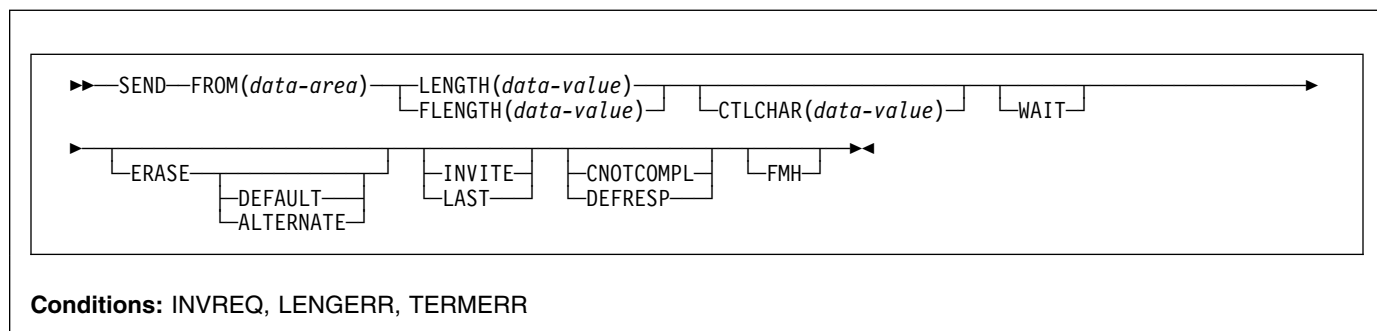


Description

SEND writes data to a terminal.

SEND (3650-3270)

Write data to a 3650 logical unit.



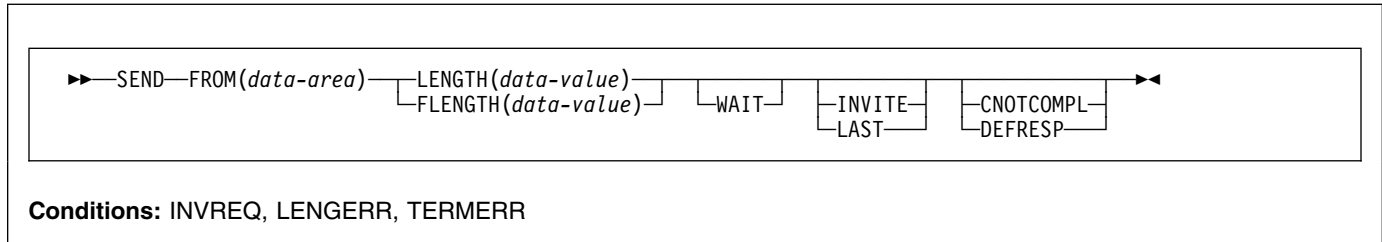
Description

SEND writes data to a terminal.

SEND (VTAM)

SEND (3650-3653)

Write data to a 3650 (3653) logical unit.

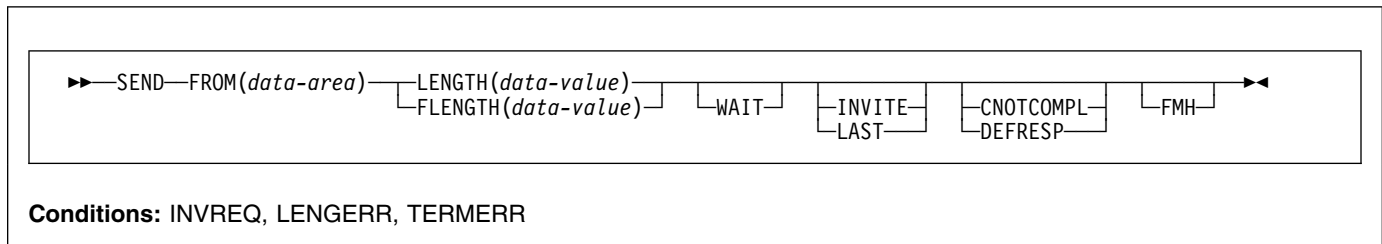


Description

SEND writes data to a terminal.

SEND (3650-3680)

Write data to a 3650 (3680) logical unit.



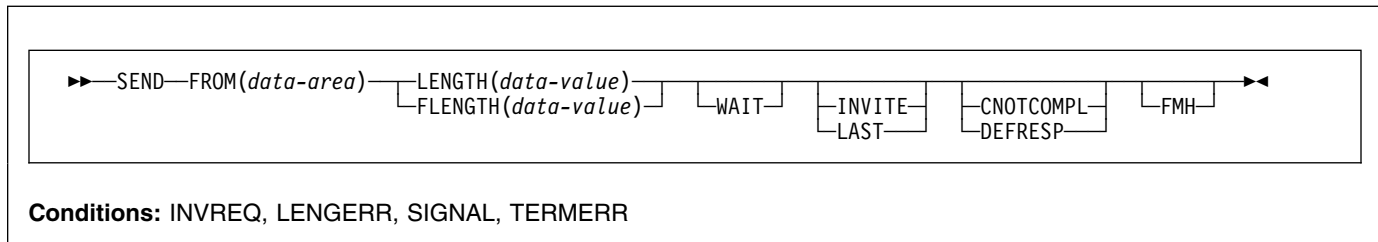
Description

SEND writes data to a terminal.

SEND (VTAM)

SEND (3790 full-function or inquiry)

Write data to a 3790 full-function or inquiry logical unit.



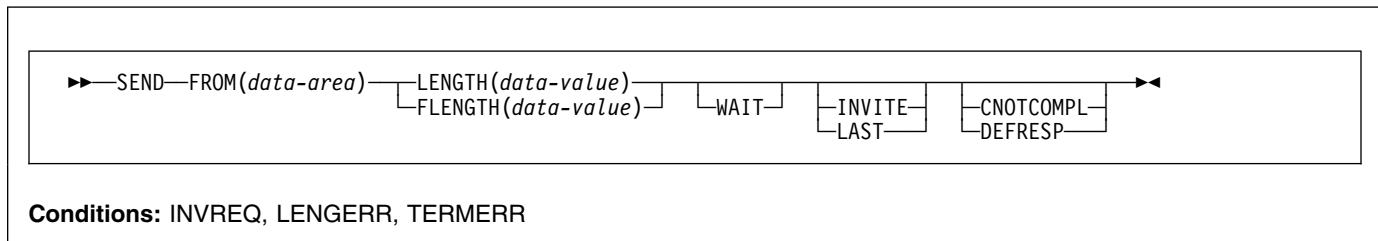
Description

SEND writes data to a terminal. This form of SEND also applies to the following devices:

- 3650/3680 full-function logical unit
- 3770 full-function logical unit

SEND (3790 SCS)

Write data to a 3790 SCS printer logical unit.

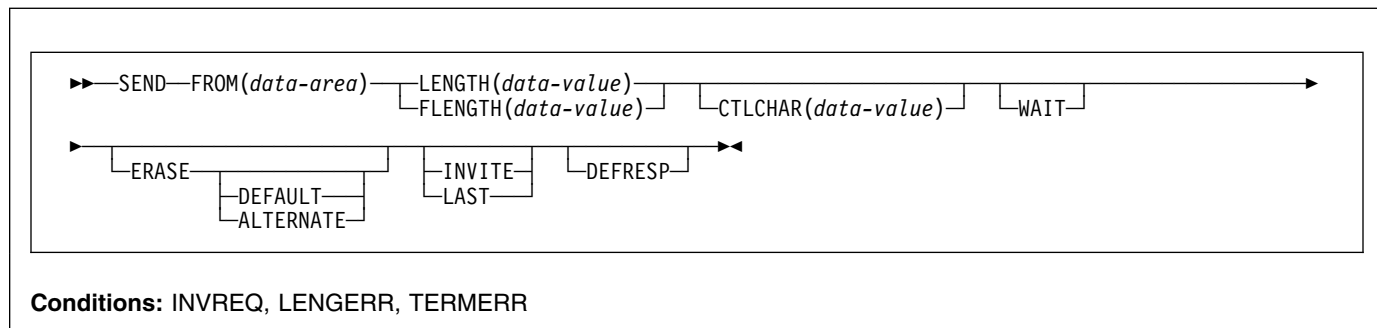


Description

SEND writes data to a terminal.

SEND (3790 3270-display)

Write data to a 3790 (3270-display) logical unit.

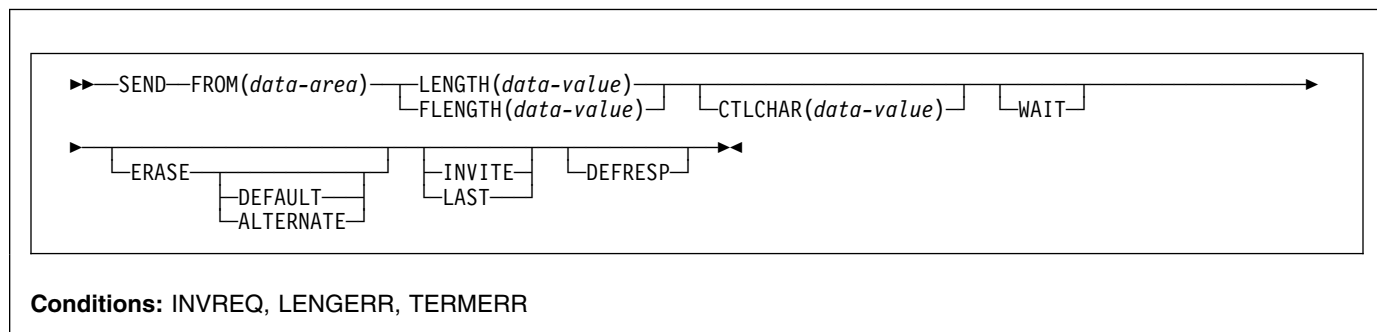


Description

SEND writes data to a terminal.

SEND (3790 3270-printer)

Write data to a 3790 (3270-printer) logical unit.



Description

SEND writes data to a terminal.

SEND (VTAM)

Options

Note: CICS Transaction Server for VSE/ESA, unlike CICS for VSE/ESA, does not support the DEST option. If a program with this option is translated on VSE/ESA, a warning message DFH7275 is produced and the option is ignored.

ALTERNATE

sets the terminal to use the ALTERNATE screen size.

ATTACHID(*name*)

specifies that an attach header (created by a BUILD ATTACH command) is to precede, and be concatenated with, the user data supplied in the FROM option. "name" (1–8 characters) identifies the attach header control block to be used in the local task.

CNOTCOMPL

indicates that the request/response unit (RU) sent as a result of this SEND command does not complete the chain. If this option is omitted and chain assembly has been specified, the RU terminates the chain.

CONFIRM

indicates that an application using a synchronization level 1 or 2 conversation requires a response from the remote application. A remote CICS application can respond positively by executing an ISSUE CONFIRMATION command, or negatively, by executing an ISSUE ERROR command, in which case the sending application has EIBERR and EIBERRCD set. CICS does not return control to the sending application until the response is received.

CONVID(*name*)

identifies the conversation to which the command relates. The 4-character name identifies either the token returned by a previously executed ALLOCATE command in EIBRSRCE in the EIB, or the token representing the principal session (returned by a previously executed ASSIGN command).

For compatibility with earlier releases, SESSION is accepted as a synonym for CONVID. New programs should use CONVID.

If this option is omitted, the principal facility is assumed.

CTLCHAR(*data-value*)

specifies a 1-byte write control character (WCC†) that controls a SEND command for a 3270. A COBOL user must specify a data area containing this character. If the option is omitted, all modified data tags are reset to zero and the keyboard is restored.

DEFAULT

sets the terminal to use the DEFAULT screen size.

DEFRESP

indicates that a definite response is required when the output operation has been completed.

ERASE

specifies that the screen printer buffer or partition is to be erased and the cursor returned to the upper left corner of the screen. (This option applies only to the 3270, or 8775, and to the 3604 Keyboard Display.)

The first output operation in any transaction, or in a series of pseudoconversational transactions, should always specify ERASE. For transactions attached to 3270 screens or printers, unless explicitly overridden by the DEFAULT or ALTERNATE option, this also ensures that the correct screen size is selected, as defined for the transaction by the SCRNSIZE option in the RDO PROFILE resource definition.

See the *CICS Resource Definition Guide* manual for further information about RDO.

FLENGTH(*data-value*)

A fullword alternative to LENGTH.

FMH

specifies that a function management header has been included in the data to be written. If the ATTACHID option is specified as well, the concatenated FMH flag is set in the attach FMH.

FROM(*data-area*)

specifies the data to be written to the logical unit, or a partner transaction.

INVITE

For the SEND (APPC) command, INVITE allows an application program to add a change-direction indicator to data already sent to a process in a connected APPC system. Control data is not transmitted by CICS until the subsequent execution of a WAIT or a SYNCPOINT command, unless CONFIRM or WAIT is also coded on the GDS SEND INVITE command.

For the other SEND commands, INVITE specifies that the next terminal control command executed for this facility is a RECEIVE. This allows optimal flows to occur.

LAST

specifies that this is the last SEND command for a transaction.

LDC(*name*)

specifies the 2-character mnemonic used to determine the appropriate logical device code (LDC) numeric value. The mnemonic represents an LDC entry defined by a DFHTCT TYPE=LDC macro. See the *CICS Resource Definition Guide* for further information about the DFHTCT macro.

† Documented in the *IBM 3270 Data Stream Programmer's Reference*.

LENGTH(*data-value*)

specifies the length, as a halfword binary value, of the data to be written. For a description of a safe upper limit, see “LENGTH options” on page 5.

SESSION(*name*)

specifies the symbolic identifier (1–4 characters) of a session TCTTE. This option specifies the alternate facility to be used. If this option is omitted, the principal facility for the task is used.

STATE(*cvda*)

gets the state of the current conversation. The *cvda* values returned by CICS are:

ALLOCATED
 CONFFREE
 CONFRECEIVE
 CONFSEND
 FREE
 PENDFREE
 PENDRECEIVE
 RECEIVE
 ROLLBACK
 SEND
 SYNCFREE
 SYNCRECEIVE
 SYNCSEND

STRFIELD

specifies that the data area specified in the FROM option contains structured fields. If this option is specified, the contents of all structured fields must be handled by the application program. The CONVERSE command, rather than a SEND command, must be used if the data area contains a read partition structured field. (Structured fields are described in the *CICS 3270 Data Stream Device Guide*.)

CTLCHAR and ERASE are mutually exclusive with STRFIELD, and their use with STRFIELD generates an error message.

WAIT

specifies that processing of the command must be completed before any subsequent processing is attempted.

If the WAIT option is not specified, control is returned to the application program when processing of the command has started. A subsequent input or output request (terminal control, BMS, or batch data interchange) to the terminal associated with the task causes the application program to wait until the previous request has been completed.

Conditions

Some of the following conditions may occur in combination. If more than one occurs, only the first is passed to the application program.

EIBRCODE, however, is set to indicate all the conditions that occurred.

CBIDERR

occurs if the requested attach header control block named in ATTACHID cannot be found.

Default action: terminate the task abnormally.

IGREQCD

occurs when an attempt is made to execute a SEND command after a SIGNAL data-flow control command with a request change direction (RCD) code has been received from the logical unit.

Default action: terminate the task abnormally.

INVREQ

RESP2 values:

200 A distributed program link server application attempted to send on its function-shipping session (its principal facility).

For SEND (APPC), a distributed program link server application specified the function-shipping session (its principal facility) on the CONVID option.

also occurs (RESP2 not set) in any of the following situations:

- The CONFIRM option has been specified, but the APPC conversation is not sync level 1 or 2.
- The SEND command has been used on an APPC conversation that is not a mapped conversation or that is not using the EXEC CICS interface.

Default action: terminate the task abnormally.

LENGERR

occurs if an out-of-range value is supplied in the LENGTH or FLENGTH option.

Default action: terminate the task abnormally.

NOTALLOC

occurs if the CONVID value in the command does not relate to a conversation that is owned by the application, or if the facility specified in the command is not owned by the application.

Default action: terminate the task abnormally.

SIGNAL

occurs when an inbound SIGNAL data-flow control command has been received from a logical unit or session. EIBSIG is always set when an inbound signal is received.

Default action: ignore the condition.

SEND (VTAM)

TERMERR

occurs for a session-related error. Any action on that conversation other than a FREE causes an ATCV abend.

A CANCEL TASK request by a user node error program (NEP) may cause this condition if the task has an outstanding terminal control request active when the node abnormal condition program handles the session error.

Default action: terminate the task abnormally with abend code ATNI.

SEND (non-VTAM default)

Write data to a standard CICS terminal support.

```

  >> SEND FROM(data-area) LENGTH(data-value)
                                FLENGTH(data-value) WAIT <<
  
```

Conditions: INVREQ, LENGERR, NOTALLOC

Description

SEND writes data to a terminal. This form of the send command can be used by all CICS-supported terminals for which the other SEND descriptions are not appropriate.

SEND (MRO)

Send data on an MRO conversation.

```

  >> SEND SESSION(name) WAIT INVITE ATTACHID(name) FROM(data-area)
                                LAST
  << LENGTH(data-value) FMH DEFRESP STATE(cvda)
      FLENGTH(data-value)
  
```

Conditions: CBIDERR, INVREQ, LENGERR, NOTALLOC, TERMERR

SEND sends data to a conversation partner on an MRO conversation.

SEND (non-VTAM)

SEND (System/3)

Write data to a System/3.

Description

```
▶▶ SEND FROM(data-area) [LENGTH(data-value) | FLENGTH(data-value)] [WAIT] [ASIS] [CNOTCOMPL] ▶▶
```

Conditions: INVREQ, LENGERR

SEND writes data to a terminal. This form of the SEND command also applies to the following devices:

- 2770 data communication system
- 2780 data transmission terminal
- 3660 supermarket scanning system
- 3780 communication terminal.

SEND (System/7)

Write data to a System/7.

```
▶▶ SEND FROM(data-area) [LENGTH(data-value) | FLENGTH(data-value)] [WAIT] [PSEUDOBIN] [ASIS] ▶▶
```

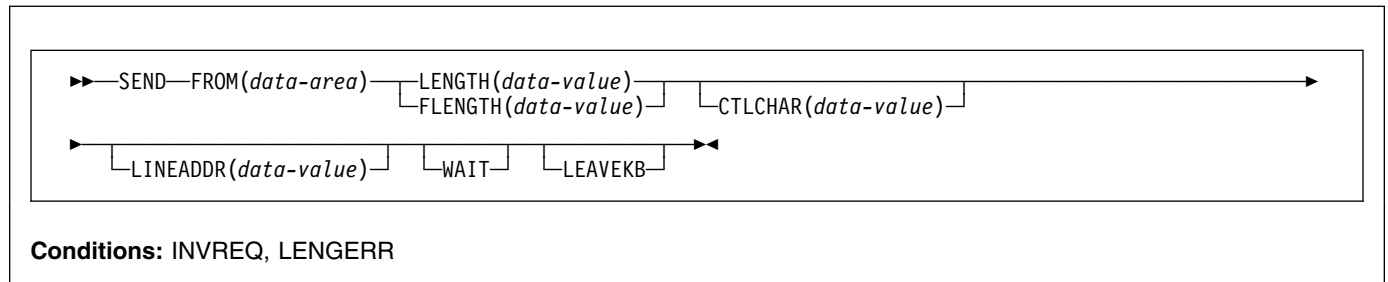
Conditions: INVREQ, LENGERR

Description

SEND writes data to a terminal. For a description of running with a System/7, see “CONVERSE (System/7)” on page 53.

SEND (2260)

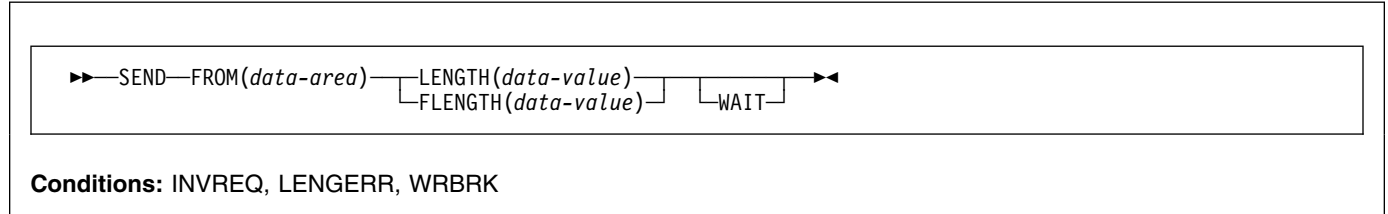
Write data to a 2260 or 2265 display station.

**Description**

SEND writes data to a terminal.

SEND (2741)

Write data to a 2741 communication terminal.

**Description**

SEND writes data to a terminal. For more information about using a 2741 communication terminal, see "CONVERSE (2741)" on page 55.

SEND (non-VTAM)

SEND (2980)

Write data to a 2980 general banking terminal system.

▶▶ SEND FROM(*data-area*) — LENGTH(*data-value*) — PASSBK —▶
 └── FLENGTH(*data-value*) ─┘ └── CBUFF ─┘

Conditions: INVREQ, LENGERR, NOPASSBKWR

Description

SEND writes data to a terminal. For more information about the 2980 general banking system, see "RECEIVE (2980)" on page 220.

SEND (3270 display)

Write data to a 3270 information display system (BTAM).

▶▶ SEND FROM(*data-area*) — LENGTH(*data-value*) — WAIT —▶
 └── FLENGTH(*data-value*) ─┘ └── ERASE ─┘ └── CTLCHAR(*data-value*) ─┘
 └── STRFIELD ─┘

Conditions: INVREQ, LENGERR

Description

SEND writes data to a terminal.

SEND (3600 BTAM)

Write data to a 3600 or 4700 finance communication system (BTAM) - only valid for remote terminals. See "BTAM programmable terminals" on page 388.

```

▶▶—SEND—FROM(data-area)—┬—LENGTH(data-value)—┬—┬—▶
                        └—FLENGTH(data-value)—┬—WAIT—┬—ASIS—┬—

```

Conditions: INVREQ, LENGERR

Description

SEND writes data to a terminal. This form of the SEND command also applies to the 4700 finance communication system. For more information about using the 3600 or 4700 finance communication system, see "CONVERSE (3600 BTAM)" on page 57.

SEND (3735)

Write data to a 3735 Programmable Buffered Terminal.

```

▶▶—SEND—FROM(data-area)—┬—LENGTH(data-value)—┬—┬—▶
                        └—FLENGTH(data-value)—┬—WAIT—┬—ASIS—┬—

```

Conditions: INVREQ, LENGERR

Description

SEND writes data to a terminal. The 3735 Programmable Buffered Terminal may be serviced by CICS in response to terminal-initiated input (automatic answering), or as a result of an automatic (automatic calling) or time-initiated transaction.

For more information about the 3735, see "CONVERSE (3735)" on page 59.

SEND (non-VTAM)

SEND (3740)

Write data to a 3740 data entry system.

```
▶▶ SEND FROM(data-area) [LENGTH(data-value)  
[FLENGTH(data-value)] [WAIT] [ASIS] ▶▶
```

Conditions: INVREQ, LENGERR

Description

SEND writes data to a terminal. In batch mode, many files are exchanged between the 3740 and CICS in a single transmission. The transmission of an input batch must be complete before an output transmission can be started.

Options

Note: CICS Transaction Server for VSE/ESA, unlike CICS for VSE/ESA, does not support the DEST option. If a program with this option is translated on VSE/ESA, a warning message DFH7275 is produced and the option is ignored.

ALTERNATE

sets the terminal to use the ALTERNATE screen size.

ASIS

indicates that output is to be sent in transparent mode (with no recognition of control characters and accepting any of the 256 possible combinations of eight bits as valid transmittable data).

Note: If you are using a katakana terminal, you might see some messages containing mixed English and katakana characters. That is because katakana terminals cannot display mixed-case output. Uppercase characters in the data stream appear as uppercase English characters, but lowercase characters appear as katakana characters. If this happens, ask your system programmer to specify MSGCASE=UPPER in the system initialization parameters so that messages contain uppercase characters only.

ATTACHID(*name*)

specifies that an attach header (created by a BUILD ATTACH command) is to precede, and be concatenated with, the user data supplied in the FROM option. "name" (1–8 characters) identifies the attach header control block to be used in the local task.

CBUFF

specifies that data is to be written to a common buffer in a 2972 control unit. The WAIT option is implied.

CNOTCOMPL

indicates that the request/response unit (RU) sent as a result of this SEND command does not complete the chain. If this option is omitted and chain assembly has been specified, the RU terminates the chain.

CTLCHAR(*data-value*)

specifies a 1-byte write control character (WCC[†]) that controls a SEND command for a 3270. A COBOL user must specify a data area containing this character. If the option is omitted, all modified data tags are reset to zero and the keyboard is restored.

DEFAULT

sets the terminal to use the DEFAULT screen size.

DEFRESP

indicates that a definite response is required when the output operation has been completed.

ERASE

specifies that the screen printer buffer or partition is to be erased and the cursor returned to the upper left corner of the screen. (This option applies only to the 3270, or 8775, and to the 3604 Keyboard Display.)

The first output operation in any transaction, or in a series of pseudoconversational transactions, should always specify ERASE. For transactions attached to 3270 screens or printers, unless explicitly overridden by the DEFAULT or ALTERNATE option, this also ensures that the correct screen size is selected, as defined for the transaction by the SCRNSIZE option in the RDO PROFILE resource definition.

FLENGTH(*data-value*)

A fullword alternative to LENGTH.

[†] Documented in the *IBM 3270 Data Stream Programmer's Reference*.

FMH

specifies that a function management header has been included in the data to be written. If the ATTACHID option is specified as well, the concatenated FMH flag is set in the attach FMH.

FROM(*data-area*)

specifies the data to be written to the logical unit or terminal.

INVITE

specifies that the next terminal control command to be executed for this facility is a RECEIVE. This allows optimal flows to occur.

LAST

specifies that this is the last output operation for a transaction and therefore the end of a bracket.

LEAVEKB

specifies that the keyboard is to remain locked at the completion of the data transfer.

LENGTH(*data-value*)

specifies the length, as a halfword binary value, of the data to be written. For a description of a safe upper limit, see "LENGTH options" on page 5.

LINEADDR(*data-value*)

specifies that the writing is to begin on a specific line of a 2260/2265 screen. The data value is a halfword binary value in the range 1 through 12 for a 2260, or 1 through 15 for a 2265.

PASSBK

specifies that communication is with a passbook. The WAIT option is implied.

PSEUDOBIN (*start-stop only*)

specifies that the data being written is to be translated from System/7 hexadecimal to pseudobinary.

SESSION(*name*)

specifies the symbolic identifier (1–4 characters) of a session TCTTE. This option specifies the alternate facility to be used. If this option is omitted, the principal facility for the task is used.

STATE(*cvda*)

gets the state of the transaction program. The cvda values returned by CICS are:

ALLOCATED
FREE
PENDFREE
RECEIVE
ROLLBACK
SEND
SYNCFREE
SYNCRECEIVE
SYNCSSEND

STRFIELD

specifies that the data area specified in the FROM option contains structured fields. If this option is specified, the contents of all structured fields must be handled by the application program. The CONVERSE command, rather than a SEND command, must be used if the data area contains a read partition structured field. (Structured fields are described in the *CICS 3270 Data Stream Device Guide*.)

CTLCHAR and ERASE are mutually exclusive with STRFIELD, and their use with STRFIELD generates an error message.

WAIT

specifies that processing of the command must be completed before any subsequent processing is attempted.

If the WAIT option is not specified, control is returned to the application program when processing of the command has started. A subsequent input or output request (terminal control, BMS, or batch data interchange) to the terminal associated with the task causes the application program to wait until the previous request has been completed.

Conditions**CBIDERR**

occurs if the requested attach header control block named in ATTACHID cannot be found.

Default action: terminate the task abnormally.

INVREQ

RESP2 values:

200 occurs if a distributed program link server application attempted to send on its function-shipping session (its principal facility)

Default action: terminate the task abnormally.

LENGERR

occurs if an out-of-range value is supplied in the LENGTH or FLENGTH option.

Default action: terminate the task abnormally.

NOPASSBKWR

occurs if no passbook is present.

Default action: terminate the task abnormally.

NOTALLOC

occurs if the facility specified in the command is not owned by the application.

Default action: terminate the task abnormally.

TERMERR

occurs for a terminal-related error, such as a session failure. This condition applies to VTAM-connected terminals only.

SEND (non-VTAM)

A CANCEL TASK request by a user node error program (NEP) may cause this condition if the task has an outstanding terminal control request active when the node abnormal condition program (DFHZNAC) handles the session error.

Default action: terminate the task abnormally withabend code ATNI.

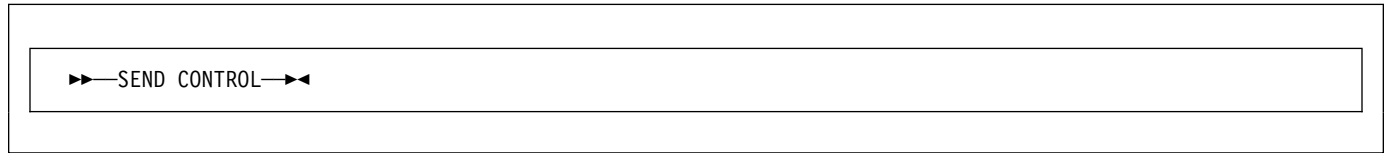
WRBRK

occurs if the command is terminated by the attention key.

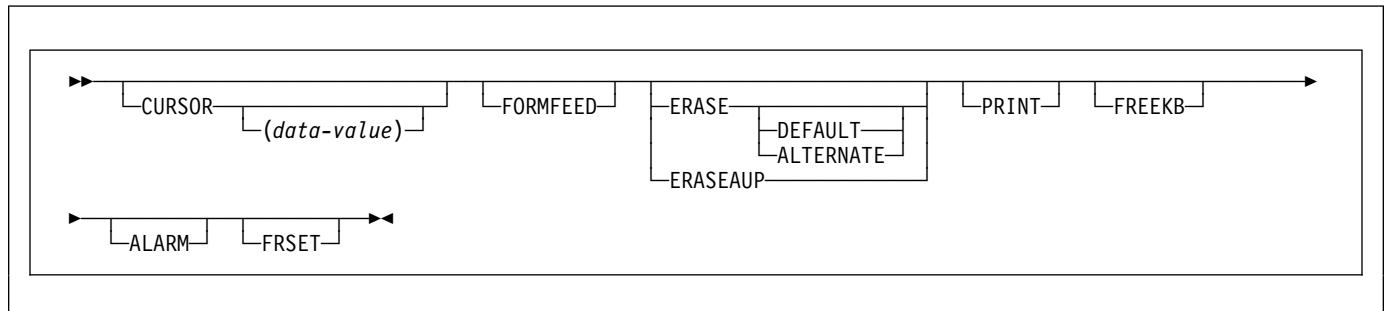
Default action: ignore the condition.

SEND CONTROL

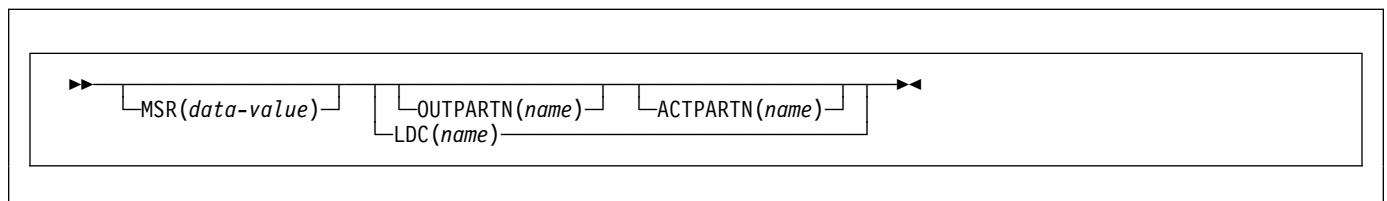
Send device controls to a terminal without map or text data. The keywords are separated into those supported by minimum, standard, and full BMS. For further information about BMS, see the *CICS Application Programming Guide*.



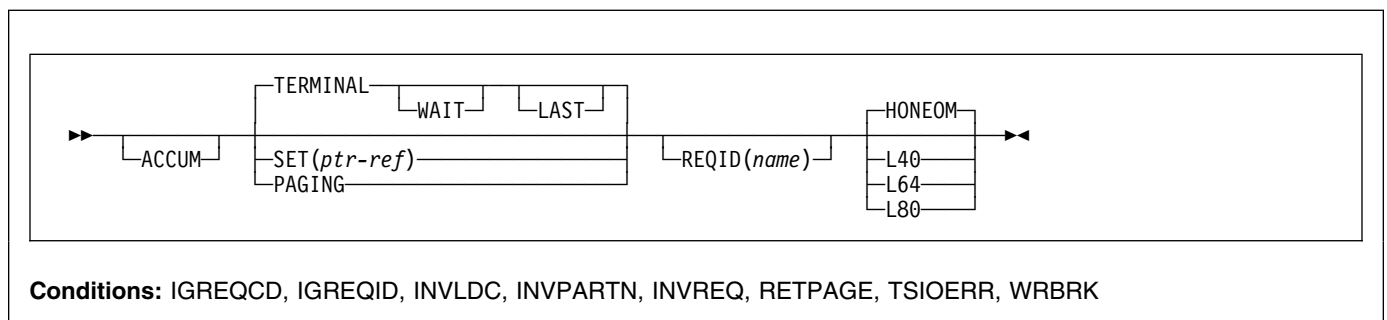
Minimum BMS:



Standard BMS:



Full BMS:



Description

SEND CONTROL sends device controls to a terminal.

When using the SEND CONTROL command with any of the ALARM, FREEKB, FRSET, HONEOM, L40, L64, L80, or PRINT options, see CTRL on page 422 for a description of the option priority.

ACCUM

specifies that this command is one of a number of commands that are used to build a logical message. The logical message is completed by a SEND PAGE command, or deleted by a PURGE MESSAGE command.

SEND CONTROL

ACTPARTN(*name*)

specifies the name (1–2 characters) of the partition to be activated. Activating a partition moves the cursor into the specified partition, and unlocks the keyboard for the specified partition.

This option is ignored if the target terminal does not support partitions, or if there is no application partition set.

ALARM

specifies that the 3270 audible alarm feature is to be activated. For logical units supporting FMHs (except interactive and batch logical units), ALARM instructs BMS to set the alarm flag in the FMH.

ALTERNATE

sets the terminal to use the ALTERNATE screen size.

CURSOR(*data-value*)

specifies the location the 3270 or 3604 cursor is returned to on completion of a SEND CONTROL command.

The data value must be a halfword binary value that specifies the cursor position relative to zero; the range of values that can be specified depends on the size of the screen being used.

If ACCUM is being used, the most recent value of CURSOR specified is used to position the cursor.

The value specified in the CURSOR option must be positive. A negative value leads to unpredictable results.

If this option is omitted, the cursor is positioned at position zero of the screen.

DEFAULT

sets the terminal to use the DEFAULT screen size.

ERASE

specifies that the screen printer buffer or partition is to be erased and the cursor returned to the upper left corner of the screen. (This option applies only to the 3270, or 8775, and to the 3604 Keyboard Display.)

The first output operation in any transaction, or in a series of pseudoconversational transactions, should always specify ERASE. For transactions attached to 3270 screens or printers, unless explicitly overridden by the DEFAULT or ALTERNATE option, this also ensures that the correct screen size is selected, as defined for the transaction by the SCRNSIZE option in the RDO PROFILE definition.

ERASEAUP

specifies that all unprotected character locations in the partition or the entire screen are to be erased. (This option applies only to the 3270 and 8775.)

FORMFEED

specifies that a new page is required. For 3270 printers and displays, the FORMFEED character is positioned at the start of the buffer. The application program must thus ensure that this buffer position is not overwritten by

map or text data. It is ignored if the target terminal does not support FORMFEED (that is, the RDO TYPETERM option FORMFEED (YES) was not specified).

FREEKB

specifies that the 3270 keyboard is to be unlocked. If FREEKB is omitted, the keyboard remains locked.

Note that the keyboard lock status is maintained separately for each partition on a terminal that supports partitions.

FRSET

specifies that the modified data tags (MDTs) of all fields currently in the 3270 (or partition) buffer are to be reset to the unmodified condition (that is, field reset).

This allows the ATTRB operand of DFHMDF for the next requested map to control the final status of fields written or rewritten in response to a BMS command, if no other attribute information has been written in the symbolic map.

HONEOM

specifies that the default printer line length is to be used. This length should be the same as that specified using the RDO TYPETERM options PAGESIZE or ALTPAGE.

LAST

specifies that this is the last output operation for a transaction and, therefore, the end of a bracket. This option applies to logical units only.

LDC(*name*)

specifies a 2-character mnemonic to be used to determine the logical device code (LDC) to be transmitted in the FMH to the logical unit. The mnemonic identifies an LDC entry defined by a DFHTCT TYPE=LDC macro. When an LDC is specified, BMS uses the device type, the page size, and the page status associated with the LDC mnemonic to format the message. These values are taken from the extended local LDC table for the LU, if it has one. If the LU has only a local (unextended) LDC table, the values are taken from the system LDC table. The numeric value of the LDC is obtained from the local LDC table, unless this is an unextended table and the value is not specified, in which case it is taken from the system table.

L40, L64, or L80

specifies the line length for a 3270 printer; a carrier return and line feed are forced after 40, 64, or 80 characters have been printed on a line. Unacceptable results are likely if this differs from the page width specified by the RDO TYPETERM options PAGESIZE or ALTPAGE.

When using the options, refer to CTRL on page 422 for a description of the option priority.

MSR(*data-value*)

specifies the 4-byte data value that controls the 10/63 magnetic stripe reader attached to an 8775 or 3643 terminal. A set of constants is provided in DFHMSRCA

to assist in setting this 4-byte area. See “Magnetic slot reader (MSR) control value constants, DFHMSRCA” on page 407 for a complete list. This option is ignored if the RDO TYPETERM option MSRCONTROL(NO) is specified.

OUTPARTN(*name*)

specifies the name (1–2 characters) of the partition to which data is to be sent. This option is ignored if the terminal does not support partitions, or if there is no application partition set associated with the terminal. If there is an application partition set, and the OUTPARTN option is omitted, data is sent to the partition named by the PARTN operand of the DFHMSD (see “DFHMSD” on page 427) or the DFHMDI (see “DFHMDI” on page 421) map definition macros. If maps are not used, or if there is no PARTN operand, the output is sent to the first partition in the partition set.

PAGING

specifies that the output data is not to be sent immediately to the terminal, but is to be placed in temporary storage and displayed in response to paging commands entered by the terminal operator.

If PAGING is specified with a REQID that is defined as recoverable in the temporary storage table (TST), CICS provides message recovery for logical messages if the task has reached a syncpoint.

PRINT

specifies that a print operation is to be started at a 3270 printer or at a 3275 with the printer adapter feature, or that data on an LUTYPE2 (3274/76 or 3790) is to be printed on a printer allocated by the controller. If this option is omitted, the data is sent to the printer buffer but is not printed.

REQID(*name*)

specifies a 2-character prefix to be used as part of a temporary storage identifier for CICS message recovery. Only one prefix can be specified for each logical message. The default prefix is **.

BMS message recovery is provided for a logical message only if the PAGING option is specified in the BMS SEND commands, and if the syncpoint has been reached.

SET(*ptr-ref*)

specifies the pointer to be set to the address of the output data.

The SET option specifies that completed pages are to be returned to the application program. The pointer is set to the address of a list of completed pages. See the description of the SET option in the section about full-function BMS in the *CICS Application Programming Guide* for more guidance on using the SET option.

If TIOAPFX=YES is specified in the map definition, the pointer returned contains the address of the TIOA prefix. The user data starts at offset X'0C' from the start of the TIOA prefix.

TERMINAL

specifies that the output data is to be sent to the terminal that originated the transaction.

WAIT

specifies that control should not be returned to the application program until the output operation has been completed.

If WAIT is not specified, control returns to the application program when the output operation has started. A subsequent input or output command (terminal control, BMS, or batch data interchange) causes the application program to wait until the previous command has been completed.

Conditions

IGREQCD

occurs when an attempt is made to execute a SEND CONTROL command after a SIGNAL data-flow control command with a request change direction (RCD) code has been received from an LUTYPE4 logical unit.

Default action: terminate the task abnormally.

IGREQID

occurs if the prefix specified in the REQID option is different from that established by a previous REQID option, or by default for this logical message—REQID (**).

Default action: terminate the task abnormally.

INVLDC

occurs if the specified LDC mnemonic is not included in the LDC list for the logical unit.

Default action: terminate the task abnormally.

INVPARTN

occurs if the specified partition is not defined in the partition set associated with the application program.

Default action: terminate the task abnormally.

INVREQ

RESP2 values:

200 A distributed program link server application attempted to send on its function-shipping session (its principal facility).

also occurs (RESP2 not set) in the following situation:

- Control information is output to the same partition or LDC as mapped data while a BMS logical message is active. If neither partitions nor LDCs are in use, control information is output to the same device as mapped data.

Default action: terminate the task abnormally.

RETPAGE

occurs if the SET option is specified and a completed page is ready for return to the application program.

SEND CONTROL

Default action: return control to the application program at the point immediately following the BMS SEND command.

TSIOERR

occurs if there is an irrecoverable temporary storage input/output error.

Default action: terminate the task abnormally.

WRBRK

occurs if the command is interrupted by the terminal operator pressing the ATTN key. It applies only to the 2741 Communication Terminal, and only if write break is supported for CICS.

Default action: ignore the condition.

SEND MAP

Send mapped output data to a terminal. The keywords are separated into those supported by minimum, standard, and full BMS. For further information about BMS, see the *CICS Application Programming Guide*.

```
▶▶ SEND MAP(name) ▶▶
```

Minimum BMS:

```
▶▶ MAPSET(name) [FROM(data-area) | MAPONLY] [DATAONLY] [LENGTH(data-value)] ▶▶
▶ CURSOR [data-value] [FORMFEED] [ERASE [DEFAULT | ALTERNATE] | ERASEAUP] [PRINT] [FREEKB] ▶
▶ ALARM [FRSET] ▶▶
```

Standard BMS:

```
▶▶ NLEOM [MSR(data-value)] [FMHPARM(name)] [OUTPARTN(name) | LDC(name) | ACTPARTN(name)] ▶▶
```

Full BMS:

```
▶▶ ACCUM [TERMINAL [WAIT] [LAST] | SET(ptr-ref) | PAGING] [REQID(name)] [NOFLUSH] [HONEOM | L40 | L64 | L80] ▶▶
```

Conditions: IGREQCD, IGREQID, INVLDC, INVMPsz, INVPARTN, INVREQ, OVERFLOW, RETPAGE, TSOERR, WRBRK

SEND MAP

Description

SEND MAP sends output data to a terminal.

When using the SEND MAP command with any of the ALARM, FREEKB, FRSET, HONEOM, L40, L64, L80, or PRINT options, see CTRL on page 422 for a description of the option priority.

See Appendix K, “BMS macro summary” on page 411 for map definition.

Options

ACCUM

specifies that this command is one of a number of commands that are used to build a logical message. The logical message is completed by a SEND PAGE command, or deleted by a PURGE MESSAGE command.

ACTPARTN(*name*)

specifies the name (1–2 characters) of the partition to be activated. Activating a partition moves the cursor into the specified partition, and unlocks the keyboard for the specified partition.

This option is ignored if the target terminal does not support partitions, or if there is no application partition set.

ALARM

specifies that the 3270 audible alarm feature is to be activated. For logical units supporting FMHs (except interactive and batch logical units), ALARM instructs BMS to set the alarm flag in the FMH.

When using the ALARM option, refer to CTRL on page 422 for a description of the option priority.

ALTERNATE

sets the terminal to use the ALTERNATE screen size.

CURSOR(*data-value*)

specifies the location to which the 3270 or 3604 cursor is to be returned upon completion of a SEND MAP command.

The data value must be a halfword binary value that specifies the cursor position relative to zero; the range of values that can be specified depends on the size of the screen being used. If no data value is specified, symbolic cursor positioning is assumed. See the section about minimum BMS in the *CICS Application Programming Guide* for more information about symbolic cursor positioning.

This option overrides any IC option of the ATTRB operand of DFHMDF. If ACCUM is being used, the most recent value of CURSOR specified is used to position the cursor.

The value specified in the CURSOR option must be positive. A negative value leads to unpredictable results.

DATAONLY

specifies that only application program data is to be written. The attribute characters (3270 only) must be specified for each field in the supplied data. If the attribute byte in the user-supplied data is set to X'00', the attribute byte on the screen is unchanged. Any default data or attributes from the map are ignored.

DEFAULT

sets the terminal to use the DEFAULT screen size.

ERASE

specifies that the screen printer buffer or partition is to be erased and the cursor returned to the upper left corner of the screen. (This option applies only to the 3270, or 8775, and to the 3604 Keyboard Display.)

The first output operation in any transaction, or in a series of pseudoconversational transactions, should always specify ERASE. For transactions attached to 3270 screens or printers, unless explicitly overridden by the DEFAULT or ALTERNATE option, this also ensures that the correct screen size is selected, as defined for the transaction by the SCRNSIZE option in the RDO PROFILE definition.

ERASEAUP

specifies that before this page of output is displayed, all unprotected character locations in the partition or the entire screen are to be erased. (This option applies only to the 3270 and 8775.)

FMHPARM(*name*)

specifies the name (1–8 characters) of the outboard map to be used. (This option applies only to 3650 logical units with outboard formatting.)

FORMFEED

specifies that a new page is required. For 3270 printers and displays, the FORMFEED character is positioned at the start of the buffer. The application program must thus ensure that this buffer position is not overwritten by map or text data. It is ignored if the target terminal does not support FORMFEED (that is, the RDO TYPETERM option FORMFEED (YES) was not specified).

FREEKB

specifies that the 3270 keyboard should be unlocked after the data is written. If FREEKB is omitted, the keyboard remains locked.

Note that the keyboard lock status is maintained separately for each partition on a terminal that supports partitions.

When using the FREEKB option, refer to CTRL on page 422 for a description of the option priority.

FROM(*data-area*)

specifies the data area containing the data to be processed. If this field is not specified, the name defaults to the name of the map suffixed with an O. This includes the 12-byte prefix generated by the

TIOAPFX=YES option on the DFHMDI and DFHMSD BMS map definitions (see pages 425 and 432).

FRSET

specifies that the modified data tags (MDTs) of all fields currently in the 3270 (or partition) buffer are to be reset to the unmodified condition (that is, field reset) before any map data is written to the buffer.

This allows the ATTRB operand of DFHMDF for the requested map to control the final status of fields written or rewritten in response to a BMS command, if no other attribute information has been written in the symbolic map.

When using the FRSET option refer to CTRL on page 422 for a description of the option priority.

HONEOM

specifies that the default printer line length is to be used. This length should be the same as that specified using the RDO TYPETERM options PAGESIZE or ALTPAGE, and the same as the printer platen width; otherwise the data may not format correctly.

When using the HONEOM option, refer to CTRL on page 422 for a description of the option priority.

LAST

specifies that this is the last output operation for a transaction and, therefore, the end of a bracket. This option applies to logical units only.

LDC(*name*)

specifies a 2-character mnemonic to be used to determine the logical device code (LDC) to be transmitted in the FMH to the logical unit. The mnemonic identifies an LDC entry defined by a DFHTCT TYPE=LDC macro.

When an LDC is specified, BMS uses the device type, the page size, and the page status associated with the LDC mnemonic to format the message. These values are taken from the extended local LDC table for the logical unit, if it has one. If the logical unit has only a local (unextended) LDC table, the values are taken from the system LDC table. The numeric value of the LDC is obtained from the local LDC table, unless this is an unextended table and the value is not specified, in which case it is taken from the system table.

If the LDC option is omitted, the LDC mnemonic specified in the DFHMSD macro is used; see "DFHMSD" on page 427. If the LDC option has also been omitted from the DFHMSD macro, the action depends on the type of logical unit, as follows:

3601 logical unit

The first entry in the local or extended local LDC table is used, if there is one. If a default cannot be obtained in this way, a null LDC numeric value (X'00') is used. The page size used is the value that is specified in the RDO TYPETERM options

PAGESIZE or ALTPAGE, or (1,40) if such a value is not specified.

LUTYPE4 logical unit, batch logical unit, or batch data interchange logical unit

The local LDC table is not used to supply a default LDC; instead, the message is directed to the logical unit console (that is, to any medium that the logical unit elects to receive such messages). For a batch data interchange logical unit, this does not imply sending an LDC in an FMH. The page size is obtained in the manner described for the 3601 logical unit.

LENGTH(*data-value*)

specifies the length of the data to be formatted as a halfword binary value.

If the data area sending the map is longer than the data to be mapped, LENGTH should be specified. This should include the length of the 12-byte prefix generated by the TIOAPFX=YES option on the DFHMDI and DFHMSD BMS map definitions (see pages 425 and 432). For a description of a safe upper limit, see "LENGTH options" on page 5.

L40, L64, or L80

specifies the line length for a 3270 printer; a carrier return and line feed are forced after 40, 64, or 80 characters have been printed on a line. Unacceptable results are likely if this differs from the page width specified by the RDO TYPETERM options PAGESIZE or ALTPAGE.

When using the options, refer to CTRL on page 422 for a description of the option priority.

MAP(*name*)

specifies the name (1–7 characters) of the map to be used.

MAPONLY

specifies that only default data from the map is to be written.

MAPSET(*name*)

specifies the unsuffixed name (1–7 characters) of the mapset to be used. The mapset must reside in the CICS program library. The mapset can be defined either by using RDO or by program autoinstall when the mapset is first used. If this option is not specified, the name given in the MAP option is assumed to be that of the mapset.

The number of maps per mapset is limited to a maximum of 9998.

MSR(*data-value*)

specifies the 4-byte data value that controls the 10/63 magnetic stripe reader attached to an 8775 or 3643 terminal. A set of constants is provided in DFHMSRCA to assist in setting this 4-byte area. See "Magnetic slot reader (MSR) control value constants, DFHMSRCA" on

SEND MAP

page 407 for a complete list. This option is ignored if the RDO TYPETERM option MSRCONTROL (NO) was specified.

NLEOM

specifies that data for a 3270 printer or a 3275 display with the printer adapter feature should be built with blanks and new-line (NL) characters, and that an end-of-message (EM) character should be placed at the end of the data. As the data is printed, each NL character causes printing to continue on the next line, and the EM character terminates printing.

This option must be specified in the first SEND MAP command used to build a logical message. The option is ignored if the device receiving the message (direct or routed) is not one of those mentioned above.

If this option is used, buffer updating and attribute modification of fields previously written into the buffer are not allowed. CICS includes the ERASE option with every write to the terminal.

The NL character occupies a buffer position. A number of buffer positions, equivalent to the value of the RDO TYPETERM options PAGESIZE or ALTPAGE, for that terminal, is unavailable for data. This may cause data to wrap around in the buffer; if this occurs, the PAGESIZE or ALTPAGE value must be reduced.

The NLEOM option overrides the ALARM option if the latter is present.

NOFLUSH

specifies that CICS does not clear pages on completion but returns control to the program (having set the OVERFLOW condition in EIBRESP).

OUTPARTN(*name*)

specifies the name (1–2 characters) of the partition to which data is to be sent. This option is ignored if the terminal does not support partitions, or if there is no application partition set associated with the terminal. If there is an application partition set, and the OUTPARTN option is omitted, data is sent to the partition named by the PARTN operand of the DFHMSD or DFHMDI map definitions. If maps are not used, or if there is no PARTN operand, the output is sent to the first partition in the partition set.

PAGING

specifies that the output data is not to be sent immediately to the terminal, but is to be placed in temporary storage and displayed in response to paging commands entered by the terminal operator.

If PAGING is specified with a REQID that is defined as recoverable in the temporary storage table (TST), CICS provides message recovery for logical messages if the task has reached a syncpoint.

PRINT

specifies that a print operation is to be started at a 3270 printer or at a 3275 with the printer adapter feature, or that data on an LUTYPE2 (3274/76 or 3790) is to be

printed on a printer allocated by the controller. If this option is omitted, the data is sent to the printer buffer but is not printed.

When using the PRINT option, refer to CTRL on page 422 for a description of the option priority.

REQID(*name*)

specifies a 2-character prefix to be used as part of a temporary storage identifier for CICS message recovery. Only one prefix can be specified for each logical message. The default prefix is **.

BMS message recovery is provided for a logical message only if the PAGING option is specified in the BMS SEND commands and if the syncpoint has been reached.

SET(*ptr-ref*)

specifies the pointer to be set to the address of the input or output data.

The SET option specifies that completed pages are to be returned to the application program. The pointer is set to the address of a list of completed pages. See the description of the SET option in the section on full BMS in the *CICS Application Programming Guide* for more guidance about using the SET option.

The application program regains control either immediately following the SEND MAP command (if the current page is not yet completed), or at the label specified in a HANDLE CONDITION RETPAGE command, if the page has been completed.

If TIOAPFX=YES is specified in the map definition, the pointer returned contains the address of the TIOA prefix. The user data starts at offset X'0C' from the start of the TIOA prefix.

TERMINAL

specifies that the output data is to be sent to the terminal that originated the transaction.

WAIT

specifies that control should not be returned to the application program until the output operation has been completed.

If WAIT is not specified, control returns to the application program when the output operation has started. A subsequent input or output command (terminal control, BMS, or batch data interchange) causes the application program to wait until the previous command has been completed.

Conditions

Some of the following conditions may occur in combination. If more than one occurs, only the first is passed to the application program.

IGREQCD

occurs when an attempt is made to execute a SEND MAP command after a SIGNAL data-flow control

command with a request change direction (RCD) code has been received from an LUTYPE4 logical unit.

Default action: terminate the task abnormally.

IGREQID

occurs if the prefix specified in the REQID option is different from that established by a previous REQID option, or by default for this logical message—REQID (**).

Default action: terminate the task abnormally.

INVLDC

occurs if the specified LDC mnemonic is not included in the LDC list for the logical unit.

Default action: terminate the task abnormally.

INVMPsz

occurs if the specified map is too wide for the terminal, or if a HANDLE CONDITION OVERFLOW command is active and the specified map is too long for the terminal.

Default action: terminate the task abnormally.

INVPARTN

occurs if the specified partition is not defined in the partition set associated with the application program.

Default action: terminate the task abnormally.

INVREQ

RESP2 values:

200 Command not allowed for a distributed program link server program.

also occurs (RESP2 not set) in any of the following situations:

- Text data is output to the same partition or LDC as mapped data while a BMS logical message is active. If neither partitions nor LDCs are in use, text data is output to the same logical message as mapped data.
- A separate SEND MAP command with the ACCUM option is issued to the terminal that originated the transaction while a routed logical message is being built.
- A SEND MAP command is issued for a map without field specifications by specifying the FROM option without the DATAONLY option.
- During overflow processing, data is sent to a different LDC from the LDC that caused page overflow.
- Partitions are in use, the OUTPARTN option has not been coded on the SEND MAP command, but the PARTN operand has been coded in the mapset definition. If the condition arises, it suggests that different versions of the mapset have different PARTN values, and that the suffix deduced for the partition is not the same as the suffix of the loaded mapset.

- A SEND MAP command with the DATAONLY option is issued with a data area, supplied by the user, that resides above the 16MB line. But the length of this data area is not longer than the TIOA prefix.

Default action: terminate the task abnormally.

OVERFLOW

occurs if the mapped data does not fit on the current page. This condition is only raised if a HANDLE CONDITION OVERFLOW command is active.

Default action: ignore the condition.

RETPAGE

occurs if the SET option is specified and a completed page is ready for return to the application program.

Default action: return control to the application program at the point immediately following the BMS SEND MAP command.

TSIOERR

occurs if there is an irrecoverable temporary storage input/output error.

Default action: terminate the task abnormally.

WRBRK

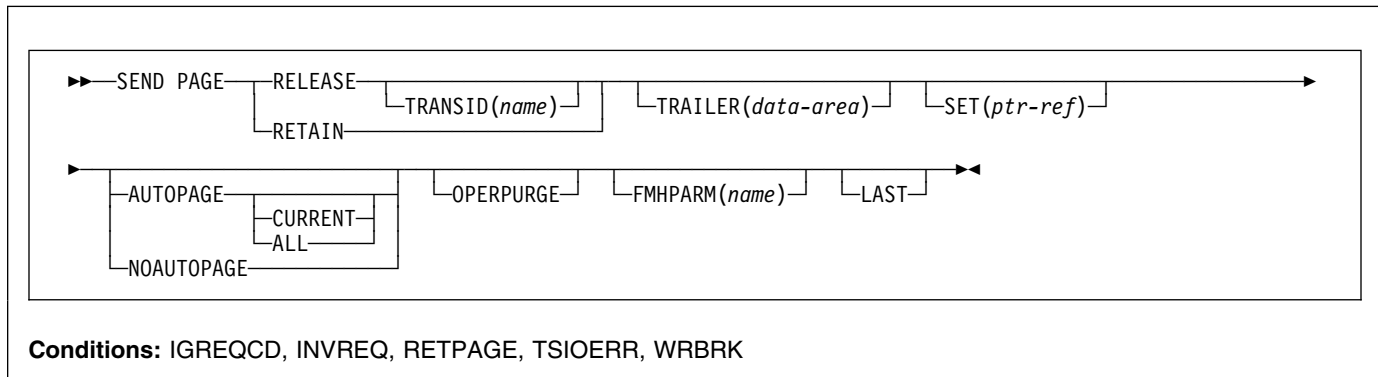
occurs if a SEND MAP command is interrupted by the terminal operator pressing the ATTN key. It applies only to the 2741 Communication Terminal, and only if write break is supported for CICS.

Default action: ignore the condition.

SEND PAGE

SEND PAGE

Send last page of data. Only supplied by full BMS. For further information about BMS, see the *CICS Application Programming Guide*.



Conditions: IGREQCD, INVREQ, RETPAGE, TSOERR, WRBRK

Description

SEND PAGE completes a BMS logical message. It causes BMS to generate a device-dependent data stream for the last (perhaps the only) page of data. Typically, this last page is only partially full.

Options can be included to specify how much control the terminal operator should have over the disposition of the logical message (AUTOPAGE, NOAUTOPAGE, and OPERPURGE), to determine whether control should return to the application program after transmission of the logical message (RELEASE or RETAIN), to add trailer data to a text logical message (TRAILER), and to return the device-dependent data stream for the last page of a logical message to the application program (SET). If this is a paging message, the last page of the logical message is transmitted to temporary storage and the terminal operator paging transaction is initiated. If it is a terminal logical message, the last page is transmitted to the terminal.

This is supported by full BMS only.

Options

ALL

specifies that if the ATTN key on a 2741 is pressed while a BMS logical message is being sent to the terminal, and the WRBRK condition is not active, transmission of the current page is to cease and no additional pages are to be transmitted. The logical message is deleted.

AUTOPAGE

specifies that each page of a BMS logical message is to be sent to the terminal as soon as it is available. If paging on request is specified for the terminal by the

RDO TYPETERM option AUTOPAGE(NO), AUTOPAGE overrides it for this logical message.

AUTOPAGE is assumed for 3270 printers; it does not apply to 3270 display terminals. If neither AUTOPAGE nor NOAUTOPAGE is specified, the terminal has the paging status specified for it using the RDO TYPETERM option AUTOPAGE.

CURRENT

specifies that if the ATTN key on a 2741 is pressed while a BMS logical message is being sent to the terminal, and the WRBRK condition is not active, transmission of the current page is to cease and transmission of the next page (if any) is to begin.

FMHPARM(name)

specifies the name (1–8 characters) of the outboard map to be used. This option applies only to 3650 logical units with outboard formatting.

LAST

specifies that this is the last output operation for a transaction and, therefore, the end of a bracket. If RELEASE is specified, LAST is assumed unless the SEND PAGE command is terminating a routing operation. This option applies to logical units only.

NOAUTOPAGE

specifies that pages of a BMS logical message are to be sent one at a time to the terminal. BMS sends the first page to the terminal when the terminal becomes available or on request of the terminal operator. Subsequent pages are sent to the terminal in response to requests from the terminal operator. (Refer to the *CICS-Supplied Transactions* manual for more information about terminal operator paging commands.)

If automatic paging is specified for the terminal by the RDO TYPETERM option AUTOPAGE(YES), NOAUTOPAGE overrides it for this logical message.

For logical units, NOAUTOPAGE applies to all pages for all LDCs in the logical message.

NOAUTOPAGE does not apply to 3270 printers.

OPERPURGE

specifies that CICS is to delete the BMS logical message only when the terminal operator requests deletion. If the option is omitted, CICS deletes the message if the operator enters data that is not a paging command.

RELEASE

specifies that, after the SEND PAGE command, control is to be returned to CICS.

RETAIN

specifies that after the SEND PAGE command, control is returned to the application program when the operator has finished displaying the pages.

SET(*ptr-ref*)

specifies the pointer to be set to the address of the output data.

The SET option specifies that the last or only page is returned to the application program. The pointer is set to the address of the current page. A list of addresses is created and, if the ROUTE command is in operation, there is an address entry for each device. If the ROUTE command is not in operation, the list contains only the one entry. See the description of the SET option in the section about full BMS in the *CICS Application Programming Guide* for more guidance on using the SET option.

The application program regains control either immediately following the SEND PAGE command (if the current page is not yet completed), or at the label specified in a HANDLE CONDITION RETPAGE command if the page has been completed.

If TIOAPFX=YES is specified in the map definition, the pointer returned contains the address of the TIOA prefix. The user data starts at offset X'0C' from the start of the TIOA prefix.

TRAILER(*data-area*)

specifies the text data area that contains trailer data to be placed at the bottom of the last page only. The format of the trailer is:

2 bytes	Binary length of the data (n)
2 bytes	Binary zero
n bytes	Data.

See the *CICS Application Programming Guide* for more information.

TRANSID(*name*)

specifies the transaction identifier (1–4 alphanumeric characters) to be used with the next input message from the terminal the task is attached to. The identifier must have been defined to CICS via a RDO TRANSACTION

resource definition. TRANSID is valid only if SEND PAGE RELEASE is specified.

If this option is specified in a program that is not at the highest logical level, the specified transaction identifier is used only if a new transaction identifier is not provided in another SEND PAGE command (or in a RETURN program control command) issued in a program at a higher logical level.

Conditions

IGREQCD

occurs when an attempt is made to execute a SEND PAGE command after a SIGNAL data-flow control command with a request change direction (RCD) code has been received from an LUTYPE4 logical unit.

Default action: terminate the task abnormally.

INVREQ

RESP2 values:

200 Command not allowed for a distributed program link server program.

also occurs (RESP2 not set) in any of the following situations:

- The disposition (TERMINAL, PAGING, or SET) of a BMS logical message is changed prior to its completion by the SEND PAGE command.
- Text data is output to the same partition or LDC as mapped data while a BMS logical message is active. If neither partitions nor LDCs are in use, text data is output to the same logical message as mapped data.
- The TRAILER option is specified when terminating a logical message built with SEND MAP commands only.
- During overflow processing data is sent to a different LDC from the LDC that caused page overflow.
- The length of the trailer is negative.

Default action: terminate the task abnormally.

RETPAGE

occurs if the SET option is specified and the last or only completed page is ready for return to the application program.

Default action: return control to the application program at the point immediately following the BMS SEND PAGE command.

TSIOERR

occurs if there is an unrecoverable temporary storage input/output error.

Default action: terminate the task abnormally.

SEND PAGE

WRBRK

occurs if the SEND PAGE command is interrupted by the terminal operator pressing the ATTN key. It applies only to the 2741 Communication Terminal, and only if write break is supported for CICS.

Default action: ignore the condition.

SEND PARTNSET

This command is available on standard and full BMS only. For further information about BMS, see the *CICS Application Programming Guide*.

```

  ▶▶ SEND PARTNSET (name) ▶▶
  
```

Conditions: INVPARTNSET, INVREQ

Description

SEND PARTNSET associates the partition set specified by the PARTNSET option with the application program. If the partition set name is omitted, the terminal is reset to the base (unpartitioned) state.

Notes:

1. A SEND PARTNSET command must not be followed immediately by a RECEIVE command. The two commands must be separated by a SEND MAP, SEND TEXT, or SEND CONTROL command, so that the partition set is sent to the terminal.
2. This command is ignored if the terminal does not support partitions.

Conditions

The following conditions may occur together. If both occur, only the first one is passed to the application program.

INVPARTNSET

occurs if the partition set named in the SEND PARTNSET command is not a valid partition set (for example, it may be a mapset).

Default action: terminate the task abnormally.

INVREQ

RESP2 values:

200 Command not allowed for a distributed program link server program.

also occurs (RESP2 not set) in the following situation:

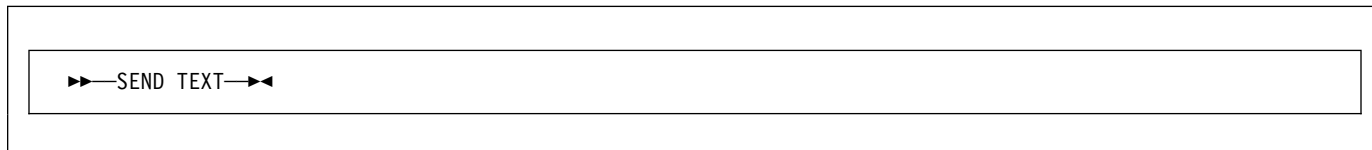
- A SEND PARTNSET command is issued while a logical message is active.

Default action: terminate the task abnormally.

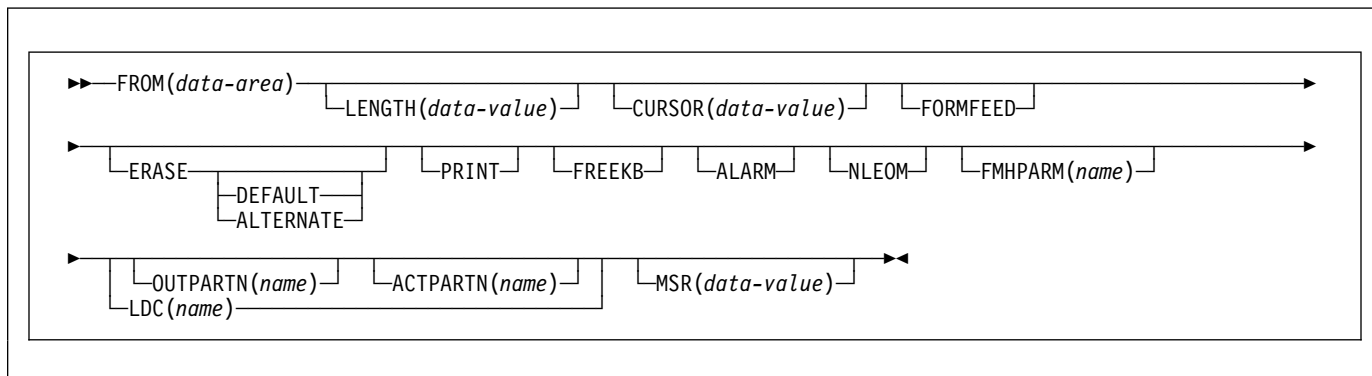
SEND TEXT

SEND TEXT

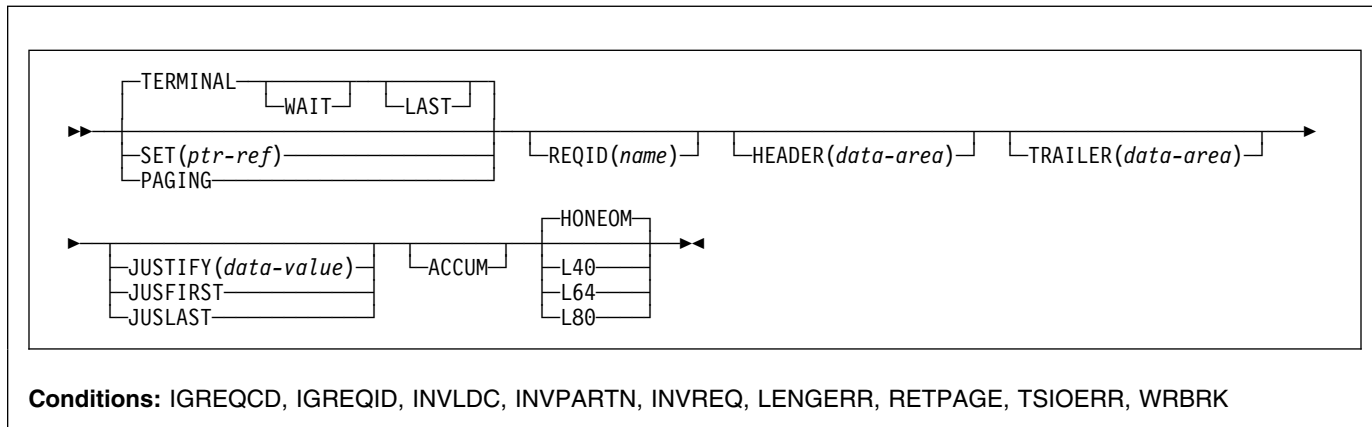
Send data without mapping. The keywords are separated into those supported by standard and full BMS. For further information about BMS, see the *CICS Application Programming Guide*.



Standard BMS:



Full BMS:



Description

SEND TEXT sends text data without mapping. The text is split into lines of the same width as the terminal, such that words are not broken across line boundaries. If the text exceeds a page, it is split into pages that fit on the terminal with application-defined headers and trailers.

When using the SEND TEXT command with any of the ALARM, FREEKB, FRSET, HONEOM, L40, L64, L80, or PRINT options, see CTRL on page 422 for a description of the option priority.

Options

ACCUM

specifies that this command is one of a number of commands that are used to build a logical message. The logical message is completed by a SEND PAGE command, or deleted by a PURGE MESSAGE command.

HEADER, JUSFIRST, JUSLAST, JUSTIFY and TRAILER all imply ACCUM.

ACTPARTN(*name*)

specifies the name (1–2 characters) of the partition to be activated. Activating a partition moves the cursor into the specified partition, and unlocks the keyboard for the specified partition.

This option is ignored if the target terminal does not support partitions, or if there is no application partition set.

ALARM

specifies that the 3270 audible alarm feature is to be activated. For logical units supporting FMHs (except interactive and batch logical units), ALARM instructs BMS to set the alarm flag in the FMH.

ALTERNATE

sets the terminal to use the ALTERNATE screen size.

CURSOR(*data-value*)

specifies the location to which the 3270 or 3604 cursor is to be returned on completion of a SEND TEXT command.

The data value must be a halfword binary value that specifies the cursor position relative to zero; the range of values that can be specified depends on the size of the screen being used.

This option overrides any IC option of the ATTRB operand of DFHMDF. If ACCUM is being used, the most recent value of CURSOR specified is used to position the cursor.

The value specified in the CURSOR option must be positive. A negative value leads to unpredictable results.

DEFAULT

sets the terminal to use the DEFAULT screen size.

ERASE

specifies that the screen printer buffer or partition is to be erased and the cursor returned to the upper left corner of the screen. (This option applies only to the 3270, or 8775, and to the 3604 Keyboard Display.)

The first output operation in any transaction, or in a series of pseudoconversational transactions, should always specify ERASE. For transactions attached to 3270 screens or printers, unless explicitly overridden by the DEFAULT or ALTERNATE option, this also ensures that the correct screen size is selected, as defined for the transaction by the SCRNSIZE option in the RDO PROFILE definition.

FMHPARM(*name*)

specifies the name (1–8 characters) of the outboard map to be used. (This option applies only to 3650 logical units with outboard formatting.)

FORMFEED

specifies that a new page is required. For 3270 printers and displays, the FORMFEED character is positioned at the start of the buffer. The application program must thus ensure that this buffer position is not overwritten by

map or text data. It is ignored if the target terminal does not support FORMFEED (that is, the RDO TYPETERM option FORMFEED(YES) was not specified).

The FORMFEED option can appear on any SEND TEXT ACCUM command. You need only specify it once within a physical page because it always forces a FORMFEED at the start of the physical page. To force a FORMFEED at the start of a particular SEND TEXT ACCUM command, use the JUSFIRST option instead.

FREEKB

specifies that the 3270 keyboard should be unlocked after the data is written. If FREEKB is omitted, the keyboard remains locked.

When using the FREEKB option, refer to CTRL on page 422 for a description of the option priority.

Note that the keyboard lock status is maintained separately for each partition on a terminal that supports partitions.

FROM(*data-area*)

specifies the data area containing the data to be sent.

HEADER(*data-area*)

specifies the header data to be placed at the beginning of each page of text data. The format of the header is:

2 bytes	Binary length of the data (n).
1 byte	Page numbering required or not (blank).
1 byte	Reserved field.
n bytes	Data.

See the *CICS Application Programming Guide* for more information.

HONEOM

specifies that the default printer line length is to be used. This length should be the same as that specified using the RDO TYPETERM options PAGESIZE or ALTPAGE, and the same as the printer platen width; otherwise the data may not format correctly.

When using the HONEOM option, refer to CTRL on page 422 for a description of the option priority.

JUSFIRST

specifies that the text data is to be placed at the top of the page. Any partially formatted page from previous requests is considered to be complete. If the HEADER option is specified, the header precedes the data. See also the description of the JUSTIFY option.

JUSLAST

specifies that the text data is to be positioned at the bottom of the page. The page is considered to be complete after the request has been processed. If the TRAILER option is specified, the trailer follows the data. See also the description of the JUSTIFY option.

JUSTIFY(*data-value*)

specifies the line of the page at which the text data is to be positioned. The data value must be a halfword binary value in the range 1 through 240. Although they

SEND TEXT

may not be specified as constants, the special values -1 and -2 can be supplied dynamically to signify JUSFIRST or JUSLAST, respectively.

LAST

specifies that this is the last output operation for a transaction and, therefore, the end of a bracket. This option applies to logical units only.

LDC(*name*)

specifies a 2-character mnemonic to be used to determine the logical device code (LDC) to be transmitted in the FMH to the logical unit. The mnemonic identifies an LDC entry defined by a DFHTCT TYPE=LDC macro.

When an LDC is specified, BMS uses the device type, the page size, and the page status associated with the LDC mnemonic to format the message. These values are taken from the extended local LDC table for the logical unit, if it has one. If the logical unit has only a local (unextended) LDC table, the values are taken from the system LDC table. The numeric value of the LDC is obtained from the local LDC table, unless this is an unextended table and the value is not specified, in which case it is taken from the system table.

LENGTH(*data-value*)

specifies the length of the data to be sent as a halfword binary value. For a description of a safe upper limit, see "LENGTH options" on page 5.

L40, L64, or L80

specifies the line length for a 3270 printer; a carrier return and line feed are forced after 40, 64, or 80 characters have been printed on a line. Unacceptable results are likely if this differs from the page width specified by the RDO TYPETERM options PAGESIZE or ALTPAGE.

When using the options, refer to CTRL on page 422 for a description of the option priority.

MSR(*data-value*)

specifies the 4-byte data value that controls the 10/63 magnetic stripe reader attached to an 8775 or 3643 terminal. A set of constants is provided in DFHMSRCA to assist in setting this 4-byte area. See "Magnetic slot reader (MSR) control value constants, DFHMSRCA" on page 407 for a complete list. This option is ignored if the RDO TYPETERM option MSRCONTROL(NO) was specified.

NLEOM

specifies that data for a 3270 printer or a 3275 display with the printer adapter feature should be built with blanks and new-line (NL) characters, and that an end-of-message (EM) character should be placed at the end of the data. As the data is printed, each NL character causes printing to continue on the next line, and the EM character terminates printing.

This option must be specified in the first SEND TEXT command used to build a logical message. The option

is ignored if the device receiving the message (direct or routed) is not one of those mentioned above.

If this option is used, buffer updating and attribute modification of fields previously written into the buffer are not allowed. CICS includes the ERASE option with every write to the terminal.

The NL character occupies a buffer position. A number of buffer positions, equivalent to the value of the RDO TYPETERM options PAGESIZE or ALTPAGE for that terminal, is unavailable for data. This may cause data to wrap around in the buffer; if this occurs, the PAGESIZE or ALTPAGE value must be reduced.

The NLEOM option overrides the ALARM option if the latter is present.

OUTPARTN(*name*)

specifies the name (1-2 characters) of the partition to which data is to be sent. This option is ignored if the terminal does not support partitions, or if there is no application partition set associated with the terminal. If there is an application partition set, and the OUTPARTN option is omitted, data is sent to the partition named by the PARTN operand of the DFHMSD or DFHMDI map definition. If maps are not used, or if there is no PARTN operand, the output is sent to the first partition in the partition set.

PAGING

specifies that the output data is not to be sent immediately to the terminal, but is to be placed in temporary storage and displayed in response to paging commands entered by the terminal operator.

If PAGING is specified with a REQID that is defined as recoverable in the temporary storage table (TST), CICS provides message recovery for logical messages if the task has reached a syncpoint.

PRINT

specifies that a print operation is to be started at a 3270 printer or at a 3275 with the printer adapter feature, or that data on an LUTYPE2 (3274/76 or 3790) is to be printed on a printer allocated by the controller. If this option is omitted, the data is sent to the printer buffer but is not printed.

When using the PRINT option, refer to CTRL on page 422 for a description of the option priority.

REQID(*name*)

specifies a 2-character prefix to be used as part of a temporary storage identifier for CICS message recovery. Only one prefix can be specified for each logical message. The default prefix is **.

BMS message recovery is provided for a logical message only if the PAGING option is specified in the BMS SEND commands and if the syncpoint has been reached.

SET(ptr-ref)

specifies the pointer to be set to the address of the data. It specifies that completed pages are to be returned to the application program. The pointer is set to the address of a list of completed pages. See the description of the SET option in the section full-function BMS in the *CICS Application Programming Guide* for more guidance about using the SET option.

The application program regains control either immediately following the BMS SEND command (if the current page is not yet completed), or at the label specified in a HANDLE CONDITION RETPAGE command if the page has been completed.

If TIOAPFX=YES is specified in the map definition, the pointer returned contains the address of the TIOA prefix. The user data starts at offset X'0C' from the start of the TIOA prefix.

TERMINAL

specifies that data is to be sent to the terminal that originated the transaction.

TRAILER(data-area)

specifies the text data area that contains trailer data to be placed at the bottom of each output page. The format of the trailer is:

2 bytes	Binary length of the data (n)
1 byte	Page numbering required or not (blank)
1 byte	Reserved field
n bytes	Data

See the *CICS Application Programming Guide* for more information.

WAIT

specifies that control should not be returned to the application program until the output operation has been completed.

If WAIT is not specified, control returns to the application program when the output operation has started. A subsequent input or output command (terminal control, BMS, or batch data interchange) causes the application program to wait until the previous command has been completed.

Conditions**IGREQCD**

occurs when an attempt is made to execute a SEND TEXT command after a SIGNAL data-flow control command with a request change direction (RCD) code has been received from an LUTYPE4 logical unit.

Default action: terminate the task abnormally.

IGREQID

occurs if the prefix specified in the REQID option on a BMS SEND command is different from that established by a previous REQID option, or by default for this logical message—REQID (**).

Default action: terminate the task abnormally.

INVLDC

occurs if the specified LDC mnemonic is not included in the LDC list for the logical unit.

Default action: terminate the task abnormally.

INVPARTN

occurs if the specified partition is not defined in the partition set associated with the application program.

Default action: terminate the task abnormally.

INVREQ

RESP2 values:

200 Command not allowed for a distributed program link server program.

also occurs (RESP2 not set) in any of the following situations:

- Text data is output to the same partition or LDC as mapped data while a BMS logical message is active. If neither partitions nor LDCs are in use, text data is output to the same logical message as mapped data.
- During overflow processing, data is sent to a different LDC from the LDC that caused page overflow.
- The length of a header on a SEND TEXT command is negative.
- The length of a trailer on a SEND TEXT command is negative.

Default action: terminate the task abnormally.

LENGERR

occurs if an out-of-range value is supplied in the LENGTH option.

Default action: terminate the task abnormally.

RETPAGE

occurs if the SET option is specified and a completed page is ready for return to the application program.

Default action: return control to the application program at the point immediately following the BMS SEND TEXT command.

TSIOERR

occurs if there is an irrecoverable temporary storage input/output error.

Default action: terminate the task abnormally.

WRBRK

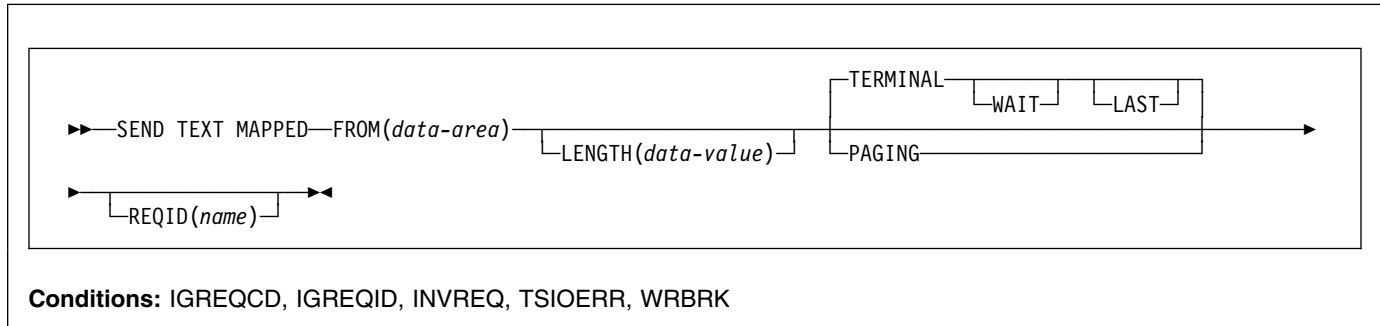
occurs if a SEND command is interrupted by the terminal operator pressing the ATTN key. It applies only to the 2741 Communication Terminal, and only if write break is supported for CICS.

Default action: ignore the condition.

SEND TEXT MAPPED

SEND TEXT MAPPED

Send data with mapping. Only supplied by full BMS. For further information about BMS, see the *CICS Application Programming Guide*.



Description

SEND TEXT MAPPED sends a page of a device-dependent data stream previously built by BMS, and returned to the application program with the SET option.

It must only be used to send data previously generated by a BMS SEND command specifying the SET option. It references a 4-byte page control area (PGA) that BMS placed at the end of the device-dependent data stream.

The length of the device-dependent data stream set in the TIOATDL field of the page buffer returned by the SET option does not include the PGA. The LENGTH option of the SEND TEXT MAPPED command should be set from this TIOATDL, and hence does not include the PGA. However, if the application program copies the page buffer returned by the SET option, it should include the PGA in the copied data.

This command is only supported by full BMS.

Options

FROM(*data-area*)

specifies the data area containing the data to be sent.

LAST

specifies that this is the last output operation for a transaction and, therefore, the end of a bracket. This option applies to logical units only.

LENGTH(*data-value*)

specifies the length of the data to be formatted as a halfword binary value. For a description of a safe upper limit, see "LENGTH options" on page 5.

PAGING

specifies that the output data is not to be sent immediately to the terminal, but is to be placed in temporary storage and displayed in response to paging commands entered by the terminal operator.

If PAGING is specified with a REQID that is defined as recoverable in the temporary storage table (TST), CICS provides message recovery for logical messages if the task has reached a syncpoint.

REQID(*name*)

specifies a 2-character prefix to be used as part of a temporary storage identifier for CICS message recovery. Only one prefix can be specified for each logical message. The default prefix is **.

BMS message recovery is provided for a logical message only if the PAGING option is specified in the BMS SEND commands and if the syncpoint has been reached.

TERMINAL

specifies that input data is to be sent to the terminal that originated the transaction.

WAIT

specifies that control should not be returned to the application program until the output operation has been completed.

If WAIT is not specified, control returns to the application program when the output operation has started. A subsequent input or output command (terminal control, BMS, or batch data interchange) causes the application program to wait until the previous command has been completed.

Conditions

IGREQCD

occurs when an attempt is made to execute a SEND TEXT command after a SIGNAL data-flow control command with a request change direction (RCD) code has been received from an LUTYPE4 logical unit.

Default action: terminate the task abnormally.

IGREQID

occurs if the prefix specified in the REQID option on a BMS SEND command is different from that established by a previous REQID option, or by default for this logical message—REQID (**).

Default action: terminate the task abnormally.

INVREQ

RESP2 values:

200 A distributed program link server application specified the function-shipping session (its principal facility) on the CONVID option.

Default action: terminate the task abnormally.

TSIOERR

occurs if there is an irrecoverable temporary storage input/output error.

Default action: terminate the task abnormally.

WRBRK

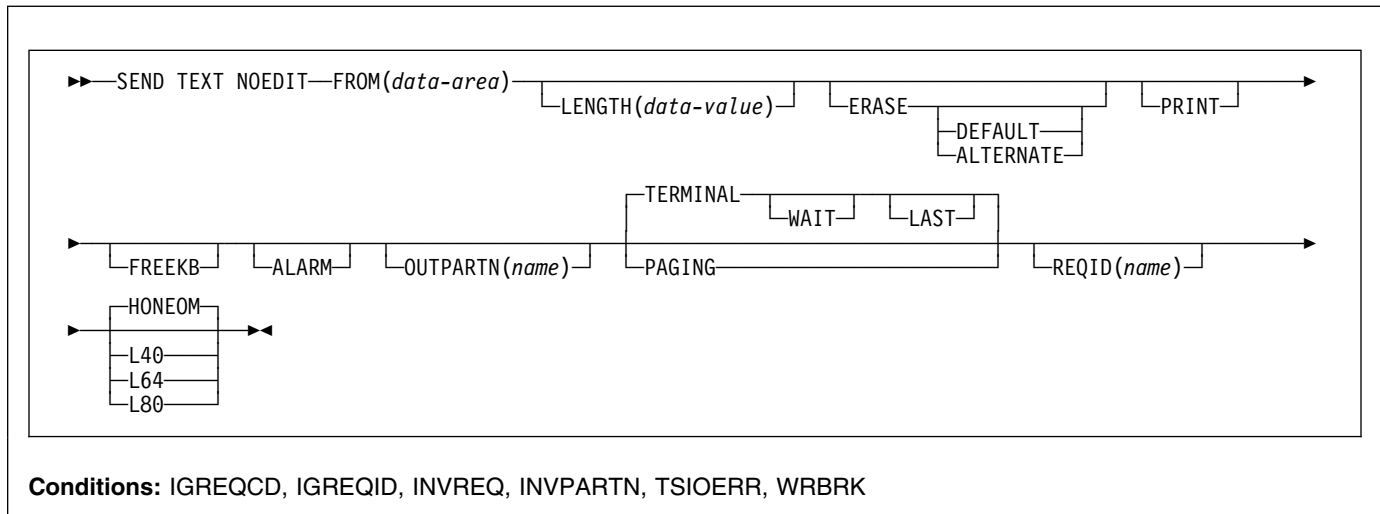
occurs if a SEND command is interrupted by the terminal operator pressing the ATTN key. It applies only to the 2741 Communication Terminal, and only if write break is supported for CICS.

Default action: ignore the condition.

SEND TEXT NOEDIT

SEND TEXT NOEDIT

Send a page. Only supplied by full BMS. For further information about BMS, see the *CICS Application Programming Guide*.



Description

SEND TEXT NOEDIT sends a page of a device-dependent data stream built by the application program. The data stream cannot contain structured fields. This command differs from a terminal control SEND, because the data stream may be written to temporary storage and interfaced to the terminal operator paging transaction (using the PAGING option). Also the device-dependent data stream can be sent to a partition (using the OUTPARTN option).

If the OUTPARTN option is specified, the data stream is sent to the specified partition. This command is used to output a user-generated data stream. It differs from a terminal control SEND in that data may be output to temporary storage (using the PAGING option), or routed like any other BMS data.

When using the SEND TEXT NOEDIT command with any of the ALARM, FREEKB, FRSET, HONEOM, L40, L64, L80, or PRINT options, see CTRL on page 422 for a description of the option priority.

This command is supported on full BMS only.

Options

ALARM

specifies that the 3270 audible alarm feature is to be activated. For logical units supporting FMHs (except interactive and batch logical units), ALARM instructs BMS to set the alarm flag in the FMH.

ALTERNATE

sets the terminal to use the ALTERNATE screen size.

DEFAULT

sets the terminal to use the DEFAULT screen size.

ERASE

specifies that the screen printer buffer or partition is to be erased and the cursor returned to the upper left corner of the screen. (This option applies only to the 3270, or 8775, and to the 3604 Keyboard Display.)

The first output operation in any transaction, or in a series of pseudoconversational transactions, should always specify ERASE. For transactions attached to 3270 screens or printers, unless explicitly overridden by the DEFAULT or ALTERNATE option, this also ensures that the correct screen size is selected, as defined for the transaction by the SCRNSIZE option in the RDO PROFILE definition.

FREEKB

specifies that the 3270 keyboard should be unlocked after the data is written. If FREEKB is omitted, the keyboard remains locked.

Note that the keyboard lock status is maintained separately for each partition on a terminal that supports partitions.

When using the FREEKB option, refer to CTRL on page 422 for a description of the option priority.

FROM(*data-area*)

specifies the data area containing the data to be sent.

HONEOM

specifies that the default printer line length is to be used. This length should be the same as that specified using the RDO TYPETERM options PAGESIZE or ALTPAGE, and the same as the printer platen width; otherwise the data may not format correctly.

When using the HONEOM option, refer to CTRL on page 422 for a description of the option priority.

LAST

specifies that this is the last output operation for a transaction and, therefore, the end of a bracket. This option applies to logical units only.

LENGTH(*data-value*)

specifies the length of the data to be sent as a halfword binary value. For a description of a safe upper limit, see "LENGTH options" on page 5.

L40, L64, or L80

specifies the line length for a 3270 printer; a carrier return and line feed are forced after 40, 64, or 80 characters have been printed on a line. Unacceptable results are likely if this differs from the page width specified by the RDO TYPETERM options PAGESIZE or ALTPAGE.

When using the options, refer to CTRL on page 422 for a description of the option priority.

OUTPARTN(*name*)

specifies the name (1–2 characters) of the partition to which data is to be sent. This option is ignored if the terminal does not support partitions, or if there is no application partition set associated with the terminal. If there is an application partition set, and the OUTPARTN option is omitted, data is sent to the partition named by the PARTN operand of the DFHMMSD or DFHMMDI map definition. If maps are not used, or if there is no PARTN operand, the output is sent to the first partition in the partition set.

PAGING

specifies that the output data is not to be sent immediately to the terminal, but is to be placed in temporary storage and displayed in response to paging commands entered by the terminal operator.

If PAGING is specified with a REQID that is defined as recoverable in the temporary storage table (TST), CICS provides message recovery for logical messages if the task has reached a syncpoint.

PRINT

specifies that a print operation is to be started at a 3270 printer or at a 3275 with the printer adapter feature, or that data on an LUTYPE2 (3274/76 or 3790) is to be printed on a printer allocated by the controller. If this option is omitted, the data is sent to the printer buffer but is not printed.

When using the PRINT option, refer to CTRL on page 422 for a description of the option priority.

REQID(*name*)

specifies a 2-character prefix to be used as part of a temporary storage identifier for CICS message recovery. Only one prefix can be specified for each logical message. The default prefix is **.

TERMINAL

specifies that the data is to be sent to the terminal that originated the transaction.

WAIT

specifies that control should not be returned to the application program until the output operation has been completed.

If WAIT is not specified, control returns to the application program when the output operation has started. A subsequent input or output command (terminal control, BMS, or batch data interchange) causes the application

SEND TEXT NOEDIT

program to wait until the previous command has been completed.

Conditions

IGREQCD

occurs when an attempt is made to execute a SEND TEXT command after a SIGNAL data-flow control command with a request change direction (RCD) code has been received from an LUTYPE4 logical unit.

Default action: terminate the task abnormally.

IGREQID

occurs if the prefix specified in the REQID option on a BMS SEND command is different from that established by a previous REQID option, or by default for this logical message—REQID (**).

Default action: terminate the task abnormally.

INVPARTN

occurs if the specified partition is not defined in the partition set associated with the application program.

Default action: terminate the task abnormally.

INVREQ

RESP2 values:

200 Command not allowed for a distributed program link server program.

also occurs (RESP2 not set) in any of the following situations:

- Text data is output to the same partition or LDC as mapped data while a BMS logical message is active. If neither partitions nor LDCs are in use, text data is output to the same logical message as mapped data.
- During overflow processing, data is sent to a different LDC from the LDC that caused page overflow.
- The length of a header on a SEND TEXT command is negative.
- The length of a trailer on a SEND TEXT command is negative.

Default action: terminate the task abnormally.

TSIOERR

occurs if there is an irrecoverable temporary storage input/output error.

Default action: terminate the task abnormally.

WRBRK

occurs if a SEND command is interrupted by the terminal operator pressing the ATTN key. It applies only to the 2741 Communication Terminal, and only if write break is supported for CICS.

Default action: ignore the condition.

SIGNOFF

Sign off from a terminal.

```
▶▶—SIGNOFF—◀◀
```

Condition: INVREQ

Description

SIGNOFF enables you to sign off from the terminal or principal facility that you previously signed on to. When sign-off is complete, the terminal reverts to the security capabilities and operator characteristics associated with the default user for this CICS region. The national language reverts to the national language of the default user, if defined, or the national language associated with the definition of the terminal.

Other resource managers

When this command is executed, CICS immediately recognizes the sign-off and establishes the default attributes for the terminal. The transaction (and any associated task-related user exits, function shipping, or distributed transaction processing) may have invoked other resource managers (RMs), for example, DL/I, DB2/VSE®, or VSAM. **It is unpredictable whether these other RMs recognize the sign-off before the transaction terminates.**

The default attributes apply for all RMs invoked by subsequent transactions at the terminal.

Conditions

INVREQ

RESP2 values:

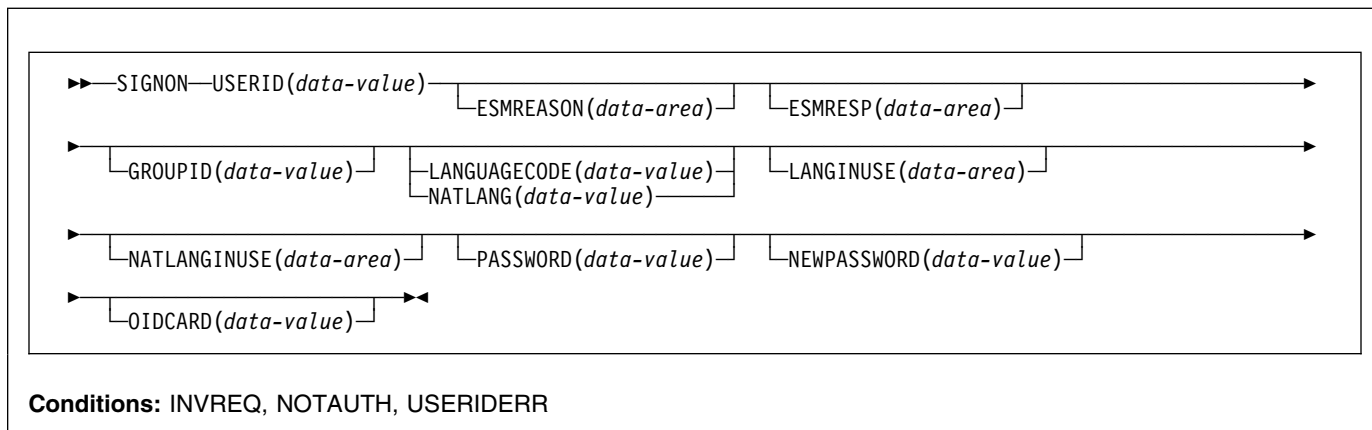
- 1 No user is currently signed on. This could be because the CICS ESM is not initialized.
- 2 There is no terminal with this task.
- 3 This task's terminal has preset security.
- 4 Sign-off is attempted using transaction routing without using the CRTE transaction.
- 18 The CICS ESM interface is not initialized.
- 200 Command not allowed for a distributed program link server program.

Default action: terminate the task abnormally.

SIGNON

SIGNON

Sign on to a terminal.



Description

SIGNON associates the security capabilities and operator characteristics of the specified user with the terminal. It allows all types of terminal, with the exception of APPC, to be signed on to. SIGNON signs on to the terminal or principal facility associated with the issuing transaction.

There is no implied sign-off with the SIGNON command. If you want to sign on a user at a terminal to which a user is already signed on, you must first issue a SIGNOFF command. Note that there is no default value for the USERID option.

PASSWORD is used as a parameter which means that if there is a dump it may become visible. You should therefore clear the field as soon as possible after using PASSWORD.

Other resource managers

When this command is executed, CICS immediately recognizes the sign-on and establishes the specified user's security and operating attributes for the terminal. The transaction (and any associated task-related user exits, function shipping, or distributed transaction processing) may have invoked other resource managers (RMs), for example, DL/I, DB2/VSE, or VSAM. **It is unpredictable whether these other RMs recognize the sign-on before the transaction terminates.**

The new user attributes apply for all RMs invoked by subsequent transactions at the terminal.

Options

If an optional input field contains all blanks, it is ignored.

ESMREASON(data-area)

returns the reason code, in a fullword binary field, that CICS receives from the external security manager.

ESMRESP(data-area)

returns the response code, in a fullword binary field, that CICS receives from the external security manager.

GROUPID(data-value)

assigns, to an ESM user group, the user that is being signed on. This overrides, for this session only, the default group name specified for the user in the ESM database.

LANGUAGECODE(data-value)

specifies the national language that the user being signed on wants CICS to use. You specify the language as a standard 3-character IBM code. This is an alternative to the 1-character code that you specify on the NATLANG option.

See Appendix I, "National language codes" on page 403 for possible values of the code.

LANGINUSE(data-area)

the LANGINUSE option allows an application program to receive the national language chosen by the sign-on process. The language is identified as a standard three-character IBM code, instead of the one-character code used by NATLANGINUSE. It is an alternative to the NATLANGINUSE option.

See Appendix I, "National language codes" on page 403 for possible values of the code.

NATLANG(*data-value*)

specifies a 1-character field identifying the national language the user wants to use during the signed-on session.

See Appendix I, “National language codes” on page 403 for possible values of the code.

NATLANGINUSE(*data-area*)

specifies a one character the national language used during the signed-on session. NATLANGINUSE corresponds to the following (in order of decreasing priority):

- The language supplied in the NATLANG option of the SIGNON command.
- The language associated with the user. This is specified in the ESM language segment.
- The language associated with the definition of the terminal.
- The language associated with the default USERID for the CICS region.
- The default language specified in the system initialization parameters.

See Appendix I, “National language codes” on page 403 for possible values of the code.

NEWPASSWORD(*data-value*)

specifies an optional 8-byte field defining a new password. This option is only valid if PASSWORD is also specified.

OIDCARD(*data-value*)

specifies an optional 65-byte field containing further security data from a magnetic strip reader (MSR) on 32xx devices.

PASSWORD(*data-value*)

specifies an 8-byte password required by the external security manager (ESM).

USERID(*data-value*)

specifies the 8-byte sign-on USERID.

Conditions**INVREQ**

RESP2 values:

- 9** The terminal is already signed on.
- 10** No terminal is associated with this task.
- 11** This task’s terminal has preset security.
- 12** The response from CICS security modules is unrecognized.
- 13** There is an unknown return code in ESMRESP from the external security manager; or the external security manager (ESM) is not active, or has failed in an unexpected way.

- 14** The required national language is not available.
- 15** Signon was attempted using transaction routing without using the CRTE transaction.
- 18** The CICS ESM interface is not initialized (SEC=NO specified as a system initialization parameter).
- 25** The terminal is of an invalid type.
- 27** The external security manager (ESM) is not active.
- 28** The required national language is invalid.
- 29** The user is already signed on. This relates to the sign-on scope checking.
- 200** Command not allowed for a distributed program link server program.

Default action: terminate the task abnormally.

NOTAUTH

RESP2 values:

- 1** A password is required.
- 2** The supplied password is wrong.
- 3** A new password is required.
- 4** The new password is not acceptable.
- 5** An OIDCARD is required.
- 6** The supplied OIDCARD is wrong.
- 16** The USERID is not authorized to use this terminal.
- 17** The USERID is not authorized to use the application.
- 19** The USERID is revoked.
- 20** The USERID’s access to the specified group has been revoked.
- 21** The sign-on failed during SECLABEL checking.
- 22** The sign-on failed because the ESM is not currently accepting sign-on.
- 23** The GROUPID is not known to the ESM.
- 24** The USERID is not contained in the GROUPID.

Default action: terminate the task abnormally.

USERIDERR

RESP2 values:

- 8** The USERID is not known to the external security manager.
- 30** The USERID is all blanks or nulls.

Default action: terminate the task abnormally.

SPOOLCLOSE

Close a report.

```

  >> SPOOLCLOSE-TOKEN(data-area) [KEEP/DELETE] <<
  
```

Conditions: INVREQ, IOERR, NOSPOOL

Description

This command closes a spool report and optionally changes its retention characteristics. You must use the NOHANDLE option, otherwise you get an APST abend.

Options

KEEP or DELETE

specifies the disposition code.

If an INPUT record is closed with disposition KEEP, it is read again when SPOOLOPEN INPUT is next issued. Closing it with DELETE results in the next report being read on the subsequent OPEN INPUT. KEEP is the default if the user fails to close the report. However if the SPOOLCLOSE command is issued without a KEEP or DELETE option, then DELETE is assumed for input reports.

For an OUTPUT report, the KEEP option is ignored. The DELETE option purges a report. When neither option is specified, the report goes to its destination node.

TOKEN(*data-area*)

This is the CICS-allocated token used to identify a report. It is an 8-byte value aligned on a fullword boundary. It is a receiver on SPOOLOPEN and a sender on all other commands.

Conditions

INVREQ

RESP2 values:

- 1 Unsupported language (RPG II).
- 4 Attempted SPOOLCLOSE without any previous valid SPOOLOPEN.
- 5 Attempted SPOOLCLOSE with a token that does not match a current token.

Default action: terminate the task abnormally.

IOERR

Indicates an input/output error. The RESP2 codes show the decimal representation of the hexadecimal POWER return and feedback codes documented in the PWRSPPL copybook. For example, a RESP2 value of 2057 is equivalent to a POWER return code of X'0809', which means "INVALID USER/REMOTE ID." of a currently open spool file.

Default action: terminate the task abnormally.

NOSPOOL

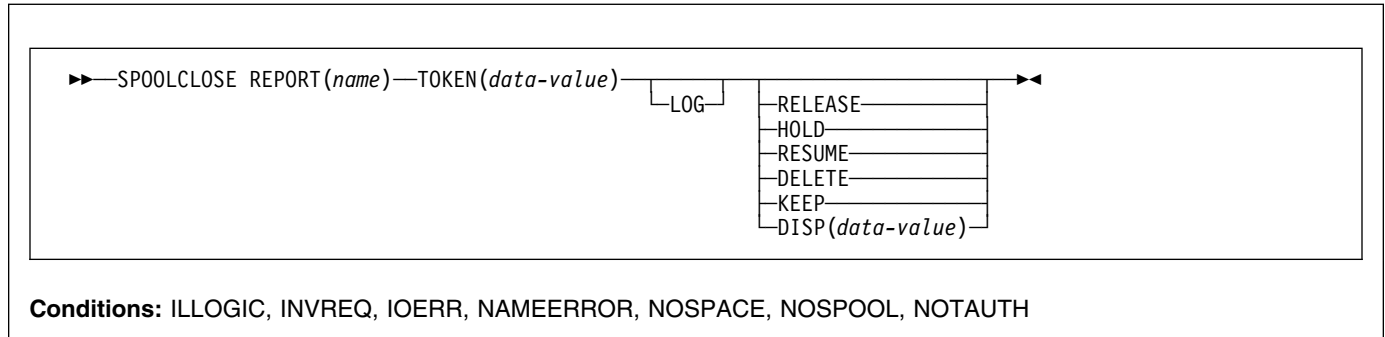
RESP2 values:

- 1 The VSE/POWER interface is not available. No subsystem is present.
- 2 The VSE/POWER interface is not available. No report controller interface is present.

Default action: terminate the task abnormally.

SPOOLCLOSE REPORT

Close a report.



Description

SPOOLCLOSE REPORT is a function of the report controller facility. For more information about the report controller, see the *CICS Report Controller User's Guide* and the *CICS Report Controller Planning Guide*.

Options

DELETE

specifies that the report can be deleted without printing; it is no longer required.

DISP(*data-value*)

specifies the POWER disposition status of the report after it has been closed (D, H, K, or L).

If you do not specify a disposition for the report, it is closed with a disposition of D.

DISP('D')

(POWER DISP=D). Specifies that, after completion of the SPOOLCLOSE REPORT command, this report should be printed. The report is set to READY status.

This option is synonymous with the RELEASE option, which is retained for compatibility with earlier releases of CICS.

DISP('H')

(POWER DISP=H). The report is created in HOLD status. After printing, it is automatically deleted.

DISP('K')

(POWER DISP=K). The report is created for immediate printing. After printing, it is retained on the POWER queue (for example, for security purposes).

This option is synonymous with the KEEP option, which is retained for compatibility with earlier releases of CICS.

DISP('L')

(POWER DISP=L). The report is created and is held on the queue until manually released. No attempt is made to print the report until the status of the report is changed by the CEMS or CEOS transactions, or by a VSE/POWER command. Once printed, the report is released and returned to the held status.

This option is synonymous with the HOLD option, which is retained for compatibility with earlier releases of CICS.

HOLD

(POWER DISP=L). Specifies that the report is to be put in HOLD status after being closed. No attempt is made to print the report until the status of the report has been changed by the CEMS or CEOS transactions, or by a POWER command.

This option is retained for compatibility with earlier releases of CICS. It is equivalent to the new option, DISP('L').

KEEP

(POWER DISP=K). Specifies that the report is created for immediate printing. After printing, it is retained on the POWER queue (for example, for security purposes).

This option is retained for compatibility with earlier releases of CICS. It is equivalent to the new option, DISP('K').

LOG

specifies that a log report is to be closed. A log report is defined as a report that can be written to by a number of applications concurrently. A log report stays open until it is closed explicitly, or it is closed implicitly at CICS termination.

A log report can be in any format.

SPOOLCLOSE REPORT

RELEASE

(POWER DISP=D). Specifies that, after completion of the SPOOLCLOSE REPORT command this report should be printed. The spooler sets the report to READY status.

The report may be printed when the specified destination becomes available.

This option is retained for compatibility with earlier releases of CICS. It is equivalent to the new option, DISP('D').

REPORT(name)

specifies the name (2–8 characters) of the report. This option is required for every command. The name must match the name specified in the SPOOLOPEN that used the same token.

The valid characters that can be used in the report name are as follows:

A through Z
0 through 9
- . / @ \$ #

The first character must not be numeric.

Do **not** use any of the following names, because you will not be able to manipulate them using CEMS or CEOS, or using POWER operator commands:

ALL
DEV
FREE
HOLD
LOCAL
PNET
RJE
STATUS.

RESUME

(POWER DISP=A). Specifies that the report being closed is to be set in appendable status. It can be opened again later to add more report records.

You should issue a SPOOLCLOSE RESUME command before you issue a SYNCPOINT, and a SPOOLOPEN RESUME after the SYNCPOINT. Otherwise, the SYNCPOINT results in the report being closed in READY status. However, there could be a transaction failure after the report has been closed, and before the SYNCPOINT. This could result in, for instance, file updates being backed out, while the report of file updates is not backed out.

TOKEN(data-value)

used by CICS to identify the report during creation. You must specify an 8-character area that contains the value returned after SPOOLOPEN REPORT (except for LOG reports, when a token area must be supplied but the returned value is not required.) The token is returned to the user to identify the report when it is reopened using a SPOOLCLOSE RESUME command.

Conditions

ILLOGIC

RESP2 values:

- 3 Incorrect report name
- 16 Incorrect token
- 17 Invalid DISP value
- 18 Incorrect userdata field
- 21 Unable to get storage for LOAD (GETVIS).

Default action: terminate the task abnormally.

INVREQ

RESP2 values:

- 4 A SPOOLCLOSE REPORT has been attempted without a preceding valid SPOOLOPEN REPORT
- 5 Token specified on SPOOLCLOSE REPORT does not match the token received on SPOOLOPEN REPORT for a given report name
- 17 A SPOOLCLOSE RESUME has been attempted for a report previously opened as a JCL type
- 18 A SPOOLCLOSE RESUME has been attempted for a report that is on the XMT queue (NODE specified on SPOOLOPEN REPORT)
- 22 Unsupported language (RPGII).

Default action: terminate the task abnormally.

IOERR

occurs when an error response is returned from POWER. RESP2 contains the POWER return and feedback codes in the format **rrffvvvv** where:

rr IJBXRETC.

ff IJBXREAS.

vv offset 4 from IJBXRUSR (PXPRETCD).

vv offset 5 from IJBXRUSR (PXPFBKCD).

IJBXRETC and IJBXREAS are detailed in the DSECT generated by MAPXPCCB.

PXPRETCD and PXPFBKCD are detailed in the DSECT PXPUSER generated as part of PWRSPLE TYPE=MAP.

Default action: terminate the task abnormally.

NAMEERROR

occurs when an attempt is made to close a report that does not exist.

Default action: terminate the task abnormally.

NOSPACE

occurs when the spool files are 90% full. If you continue creating reports without providing more space, IOERR occurs, and you will be unable to close your report.

Default action: terminate the task abnormally.

NOSPOOL

occurs when part, or all, of the POWER interface is unavailable.

RESP2 values:

- 1 No subsystem present
- 2 No report controller interface present
- 6 Interface being terminated
- 7 Interface not active
- 8 Interface being disabled
- 12 Interface has been stopped

Default action: terminate the task abnormally.

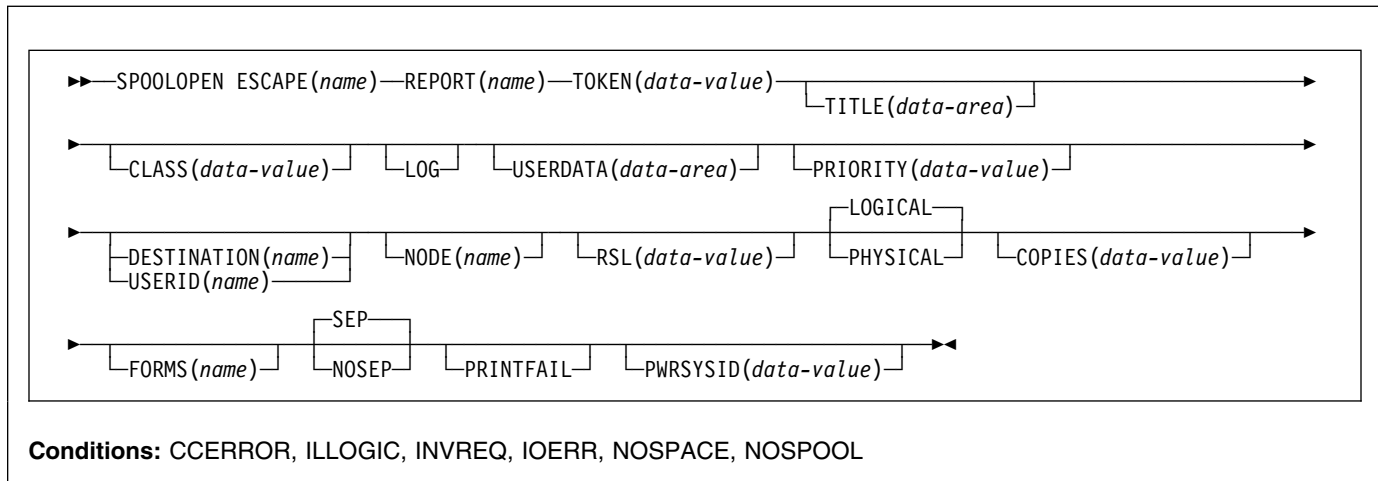
NOTAUTH

occurs when a resource security check has failed. The reasons for the failure are the same as for abend code AEY7, as listed in the *VSE/ESA Messages and Codes Volume 3* manual.

Default action: terminate the task abnormally.

SPOOLOPEN ESCAPE

Open ESC format report.



Description

SPOOLOPEN ESCAPE is a function of the report controller facility. For more information about the report controller, see the *CICS Report Controller User's Guide* and the *CICS Report Controller Planning Guide*.

Options

CLASS(data-value)

specifies the POWER output class (1-character A–Z, or numeral 0–9) for reports in all formats except JCL. The default class is specified by the SPOOL system initialization parameter.

COPIES(data-value)

specifies the number of copies (1–255) requested for this report. If not specified, 1 is assumed.

DESTINATION(name)

specifies the destination name (1–8 characters) at which the report is to be printed.

The valid characters that can be used in the destination name are as follows:

A through Z
0 through 9
@ \$
**SYSPRT

The first character must not be numeric.

Note: To avoid conflict with some POWER commands, DESTINATION should not be specified as LOCAL or SYSTEM.

ESCAPE(name)

specifies the name (1–8 characters) of a user-written escape program that is invoked at printing time to format the output.

The valid characters that can be used in the escape name are as follows:

A through Z
0 through 9
@ \$

The first character must not be numeric.

Note: If the escape routine uses CICS commands, EIBRCODE must be set to zero at exit from the escape routine.

If you use the master terminal transaction (CEMT) to disable or newcopy escape programs after a printer has been started, the effects of that CEMT command may not be felt until after the printer has been stopped.

FORMS(name)

specifies the name (1–4 characters) of the stationery to be used to print the report. At print time, a match is made between the forms specified for the report and the forms type for the printer. (The forms type for a printer can be specified/alterd using the CEMS and CEOS transactions.) If they match, the report is printed; otherwise, the report remains “in use” until the correct forms type is selected.

Note: It must be stressed that no physical check is made regarding the size of the paper in the printer; if the forms type is right, the report is printed, regardless of what paper is actually loaded.

The valid characters that can be used in the forms name are as follows:

A through Z
0 through 9
- . / @ \$

LOG

specifies that a log report is to be opened. A log report is defined as a report that can be written to by a number of applications concurrently. A log report stays open until it is closed explicitly, or it is closed implicitly at CICS termination.

A log report can be in any format.

LOGICAL

specifies that the report has logical recovery.

Reports are not committed until a POWER checkpoint is issued. With LOGICAL recovery, a POWER checkpoint is issued only when the report is closed, either by a SPOOLCLOSE REPORT command, or when a syncpoint is issued.

If an abnormal termination occurs while a task is active, the report is backed out to the last POWER checkpoint.

Logical recovery is not supported for LOG reports.

NODE(name)

specifies the (1-8 characters) name of the target node, to which the file is to be routed by the system spooler (POWER). If you specify NODE and USERID, the file may be routed to a CICS printer. If you specify NODE alone, the file may be routed only to a system printer.

The valid characters that can be used in the node name are as follows:

A through Z
0 through 9
@ \$

The first character must not be numeric.

NOSEP

specifies that no separator pages are to appear with the printed report.

PHYSICAL

specifies that this report is to have physical recovery. When creating a report with physical recovery each SPOOLWRITE command, when completed, results in a physical write to disk of all data supplied in the command (with a POWER checkpoint). Physical recovery makes heavy use of input/output and should be carefully considered before being selected.

PRINTFAIL

specifies that if CEPW abends (due to a CICS abend) before a report that is being printed has been closed, when CEPW recovers, it does not attempt to reprint this report. The report status is set to ERRPRT. Operator intervention is required before printing of the report can continue.

PRIORITY(data-value)

specifies the priority (signed binary halfword) of the report. Reports destined for the same destination that are READY are printed in order of priority. When specified, the value must be in the range from 1 through 9 (where 9 is highest priority). If not specified, a value of 3 is assumed.

PWRSYSID(data-value)

specifies the 1-character VSE/POWER SYSID to be associated with this report. When specified, the value must be in the range 1 through 9. If not specified, the report is not assigned a VSE/POWER SYSID value.

REPORT(name)

specifies the name (2-8 characters) of the report. This option is required for every command.

An attempt to open a log report with the same name as one already open is rejected.

The valid characters that can be used in the report name are as follows:

A through Z
0 through 9
- . / @ \$

The first character must not be numeric.

Do **not** use any of the following names, because you will not be able to manipulate them using CEMS or CEOS, or using POWER operator commands:

ALL
DEV
FREE
HOLD
LOCAL
PNET
RJE
STATUS.

RSL(data-value)

specifies the security key (1–6 characters) that is associated with the report. The value must be in the range 1–24 or "PUBLIC" (127). If the RSL option is not specified, PUBLIC is assumed.

When the report is created, only terminal operators with the same resource level specification are able to alter its characteristics. The report can only be printed on printers with the same RSL value specified in OPERRSL.

The security key is checked at SPOOLCLOSE of a continuous report and at SPOOLOPEN RESUME. If the terminal operator cannot match the security key, the NOTAUTH condition occurs.

After JCL reports are closed, RSL is set to 1.

SEP

specifies that two separator pages should be printed preceding the report, and two following.

SPOOLOPEN ESCAPE

TITLE(*data-area*)

specifies the title to be displayed on the CEMS or CEOS report characteristics panel. The title is a 32-character field that is not subject to any restrictions or validation. The default title is the REPORT name. USERDATA defaults to the first 16 characters of TITLE.

Note that DBCS characters can be used in TITLE, but if you do not code USERDATA, and let it default to the first 16 characters of TITLE, DBCS characters are translated when they appear on the screen.

TOKEN(*data-value*)

a token is used by CICS to identify the report during creation. The option is required on all commands.

You must specify an 8-character area (that must be word-aligned) and store the value returned by CICS after a successful SPOOLOPEN.

USERDATA(*data-area*)

specifies a 16-character report description or other reference information to be displayed on the CEMS or CEOS report list panel. If this option is omitted, the first 16 printable characters of the title are used. If TITLE is also omitted, the report name is used.

Note that DBCS characters should not be used in USERDATA, because they are translated when they appear on the screen.

USERID(*name*)

specifies the userid name (1-8 characters) where the report is to be sent to.

The valid characters that can be used in the userid name are as follows:

A through Z
0 through 9
@ \$
**SYSPRT

The first character must not be numeric.

Note: To avoid conflict with some POWER commands, USERID should not be specified as LOCAL or SYSTEM.

Conditions

CCERROR

occurs when an incorrect control character has been specified.

Default action: terminate the task abnormally.

ILLOGIC

RESP2 values:

- 1 Incorrect class
- 3 Incorrect report name
- 4 Incorrect number of copies
- 6 Incorrect priority
- 7 Incorrect forms

- 8 Incorrect destination
- 9 Incorrect node
- 10 Incorrect RSL value
- 15 Incorrect USERID
- 16 Incorrect TOKEN
- 18 Incorrect userdata field
- 23 Incorrect PWRSYSID

Default action: terminate the task abnormally.

INVREQ

RESP2 values:

- 22 Unsupported language (RPGII)

Default action: terminate the task abnormally.

IOERR

occurs when an error response is returned from POWER. RESP2 contains the POWER return and feedback codes in the format **rrffvvvv** where:

rr IJBXRETC.
ff IJBXREAS.
vv offset 4 from IJBXRUSR (PXPRETCD).
vv offset 5 from IJBXRUSR (PXPFBKCD).

IJBXRETC and IJBXREAS are detailed in the DSECT generated by MAPXPCCB.

PXPRETCD and PXPFBKCD are detailed in the DSECT PXPUSER generated as part of PWRSPPL TYPE=MAP.

Default action: terminate the task abnormally.

NOSPACE

occurs when the spool files are 90% full. If you continue creating reports without providing more space, IOERR occurs, and you will be unable to close your report.

Default action: terminate the task abnormally.

NOSPOOL

occurs when part, or all, of the POWER interface is unavailable.

RESP2 values:

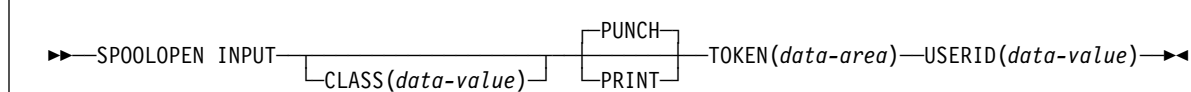
- 1 No subsystem present
- 2 No report controller interface present
- 6 Interface being terminated
- 7 Interface not active
- 8 Interface being disabled
- 12 Interface has been stopped

Default action: terminate the task abnormally.

RESP2 values:

SPOOLOPEN INPUT

Open a spool report.



Conditions: ILLOGIC, INVREQ, IOERR, NOSPACE, NOSPOOL, NOTAUTH, NOTFND

Description

This command opens a spool report for input from the system spooler to CICS. The status of the spool report must be READY (that is, its disposition must be D or K). If it is not, the response will be NOTFND.

It prepares to get (read) an existing spool data set directly using the VSE/POWER user identifier (USERID) and specified class.

When this command has been successfully executed, users should read the report and proceed to CLOSE. If SPOOLCLOSE is not issued before transaction end or syncpoint, CICS will perform an implicit SPOOLCLOSE KEEP, and will write a message to CSMT to alert the system programmer to the possible unnecessary retention of resources.

This command, if successful, will return a token, which is used later to identify the report in SPOOLREAD and SPOOLCLOSE commands. You must use the NOHANDLE option, otherwise you get an APST abend.

Options

CLASS(data-value)

is the class designation. This assigns a class to the input data set, and may be used as a selection parameter for the input report. CLASS is a 1-character sender field and is optional. If it is omitted, the default class specified by the SPOOL system initialization parameter is used. See the *CICS System Definition Guide* for more information about the SPOOL system initialization parameter.

INPUT

indicates that the report is to be read by CICS.

PRINT

specifies a listing report.

PUNCH

specifies a card image report.

TOKEN(data-area)

This is the CICS-allocated token used to identify a report. It is an 8-byte value aligned on a fullword boundary. It is a receiver on SPOOLOPEN and a sender on all other commands.

USERID(data-value)

This is the VSE/POWER identifier of the program or user that will process the report, and must be from 1–8 characters and must not include blank or null characters. The report carries this identifier and it is used to select the report at its destination. It is a sender field. For an input file, the resource security level (RSL) of the report must match the OPERRSL key of the terminal at which the transaction is running.

Conditions

ILLOGIC

RESP2 values:

- 1 Invalid class specified
- 2 Invalid PUNCH report format specified

Default action: terminate the task abnormally.

INVREQ

RESP2 values:

- 1 Unsupported language (RPG II).
- 322 Invalid USERID parameter.

Default action: terminate the task abnormally.

IOERR

Indicates an input/output error. The RESP2 codes show the decimal representation of the hexadecimal POWER return and feedback codes documented in the PWRSPPL copybook. For example, a RESP2 value of 2057 is equivalent to a POWER return code of X'0809', which means "INVALID USER/REMOTE ID." of a currently open spool file.

Default action: terminate the task abnormally.

SPOOLOPEN INPUT

NOSPACE

Indicates that there is no space on the POWER spool file to add records. If you do not have the report controller installed, spool files that are open for input are deleted by VSE/POWER. If you have the report controller installed, reports that are open for input with the LOGICAL recovery option are deleted by VSE/POWER, and reports that are open for input with the PHYSICAL recovery option are kept.

Default action: terminate the task abnormally.

NOSPOOL

RESP2 values:

- 1** The VSE/POWER interface is not available. No subsystem is present.
- 2** The VSE/POWER interface is not available. No report controller interface is present.

Default action: terminate the task abnormally.

NOTAUTH

Indicates that an attempt has been made to SPOOLOPEN INPUT a spool file without the necessary security authorization.

Default action: terminate the task abnormally.

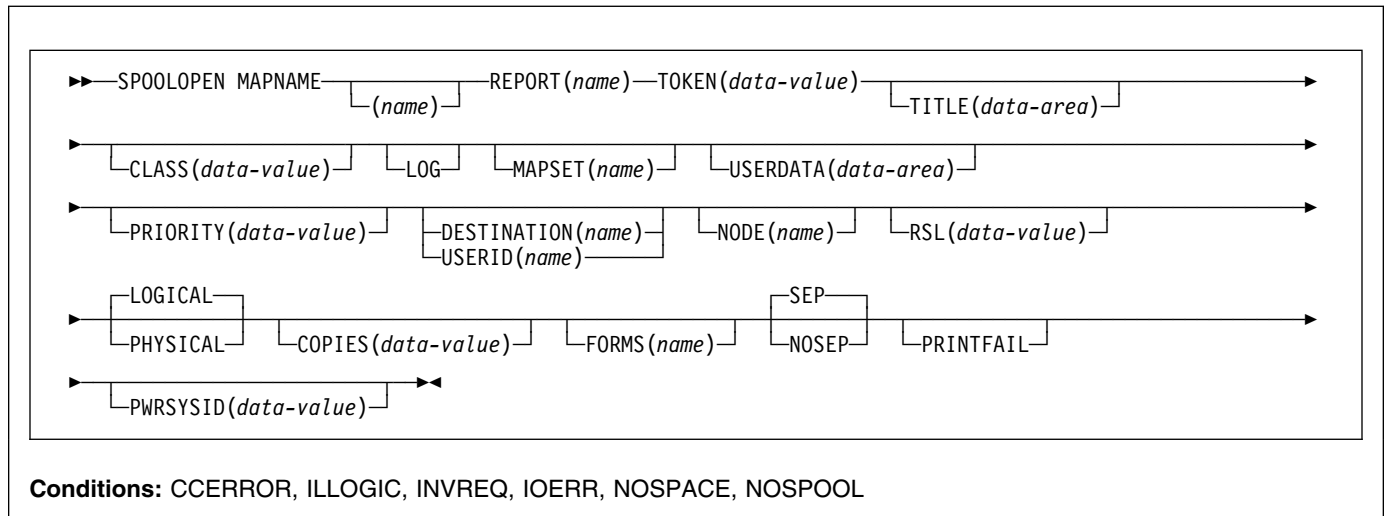
NOTFND

Indicates that no record was returned from POWER when attempting SPOOLOPEN INPUT. Ensure that the status of the spool report is READY, and that the correct queue (PRINT or PUNCH) has been specified.

Default action: terminate the task abnormally.

SPOOLOPEN MAPNAME

Open MAP format report.



Description

SPOOLOPEN MAPNAME is a function of the report controller facility. For more information about the report controller, see the *CICS Report Controller User's Guide* and the *CICS Report Controller Planning Guide*.

Options

CLASS(data-value)

specifies the POWER output class (1 character A–Z, or numeral 0–9) for reports in all formats except JCL. The default class is specified by the SPOOL system initialization parameter.

COPIES(data-value)

specifies the number of copies (signed binary halfword 1–255) requested for this report. If not specified, 1 is assumed.

DESTINATION(name)

specifies the destination name (1-8 characters) at which the report is to be printed.

The valid characters that can be used in the destination name are as follows:

A through Z
0 through 9
@ \$
**SYSPRT

The first character must not be numeric.

Note: To avoid conflict with some POWER commands, DESTINATION should not be specified as LOCAL or SYSTEM.

FORMS(name)

specifies the name (1-4 characters) of the stationery to be used to print the report. At print time, a match is made between the forms specified for the report and the forms type for the printer. (The forms type for a printer can be specified or altered using the CEMS and CEOS transactions). If they match, the report is printed; otherwise, the report remains "in use" until the correct forms type is selected.

Note: It must be stressed that no physical check is made regarding the size of the paper in the printer; if the forms type is right, the report is printed, regardless of what paper is actually loaded.

The valid characters that can be used in the forms name are as follows:

A through Z
0 through 9
- . / @ \$

LOG

specifies that a log report is to be opened. A log report is defined as a report that can be written to by a number of applications concurrently. A log report stays open until it is closed explicitly, or it is closed implicitly at CICS termination.

A log report can be in any format.

LOGICAL

specifies that the report has logical recovery.

Reports are not committed until a POWER checkpoint is issued. With LOGICAL recovery, a POWER checkpoint is issued only when the report is closed, either by a

SPOOLOPEN MAPNAME

SPOOLCLOSE command, or when a syncpoint is issued.

If an abnormal termination occurs while a task is active, the report is backed out to the last POWER checkpoint.

Logical recovery is not supported for LOG reports.

MAPNAME(*name*)

specifies the name (1-7 characters) of the default BMS map to be used for a MAP format report.

The valid characters that can be used in the destination map name are as follows:

A through Z
0 through 9
@ \$

The first character must not be numeric.

Note: Suffixes are not supported by the report controller.

MAPSET(*name*)

specifies the name (1-7 characters) of the default BMS mapset that contains the MAPNAME to be used for a MAP format report.

The valid characters that can be used in the destination mapset name are as follows:

A through Z
0 through 9
@ \$

The first character must not be numeric.

Note: Suffixes are not supported by the report controller.

NODE(*name*)

specifies the (1-8 characters) name of the target node, to which the file is to be routed by the system spooler (POWER). If you specify NODE and USERID, the file may be routed to a CICS printer. If you specify NODE alone, the file may be routed only to a system printer.

The valid characters that can be used in the node name are as follows:

A through Z
0 through 9
@ \$

The first character must not be numeric.

NOSEP

specifies that no separator pages are to appear with the printed report.

PHYSICAL

specifies that this report is to have physical recovery. When creating a report with physical recovery each SPOOLWRITE command, when completed, results in a physical write to disk of all data supplied in the command (with a POWER checkpoint). Physical

recovery makes heavy use of input/output and should be carefully considered before being selected.

PRINTFAIL

specifies that if CEPW abends (due to a CICS abend) before a report that is being printed has been closed, when CEPW recovers, it does not attempt to reprint this report. The report status is set to ERRPRT. Operator intervention is required before printing of the report can continue.

PRIORITY(*data-value*)

specifies the priority (signed binary halfword) of the report. Reports destined for the same destination that are READY are printed in order of priority. When specified, the value must be in the range from 1 through 9 (where 9 is highest priority). If not specified, a value of 3 is assumed.

PWRSYSID(*data-value*)

specifies the 1-character VSE/POWER SYSID to be associated with this report. When specified, the value must be in the range 1 through 9. If not specified, the report is not assigned a VSE/POWER SYSID value.

REPORT(*name*)

specifies the name (2-8 characters) of the report. This option is required for every command.

An attempt to open a log report with the same name as one already open is rejected.

The valid characters that can be used in the report name are as follows:

A through Z
0 through 9
- . / @ \$

The first character must not be numeric.

Do **not** use any of the following names, because you will not be able to manipulate them using CEMS or CEOS, or using POWER operator commands:

ALL
DEV
FREE
HOLD
LOCAL
PNET
RJE
STATUS.

RSL(*data-value*)

specifies the security key (1-6 characters) that is associated with the report. The value must be in the range 1-24 or "PUBLIC" (127). If the RSL option is not specified, PUBLIC is assumed.

When the report is created, only terminal operators with the same resource level specification are able to alter its characteristics. The report can only be printed on printers with the same RSL value specified in OPERRSL.

The security key is checked at SPOOLCLOSE of a continuous report and at SPOOLOPEN RESUME. If the terminal operator cannot match the security key, the NOTAUTH condition occurs.

After JCL reports are closed, RSL is set to 1.

SEP

specifies that two separator pages should be printed preceding the report, and two following.

TITLE(*data-area*)

specifies the title to be displayed on the CEMS or CEOS report characteristics panel. The title is a 32-character field that is not subject to any restrictions or validation. The default title is the REPORT name. USERDATA defaults to the first 16 characters of TITLE.

Note that DBCS characters can be used in TITLE, but if you do not code USERDATA, and let it default to the first 16 characters of TITLE, DBCS characters are translated when they appear on the screen.

TOKEN(*data-value*)

a token is used by CICS to identify the report during creation. The option is required on all commands.

You must specify an 8-character area (that must be word-aligned) and store the value returned by CICS after a successful SPOOLOPEN.

USERDATA(*data-area*)

specifies a 16-character report description or other reference information to be displayed on the CEMS or CEOS report list panel. If this option is omitted, the first 16 printable characters of the title are used. If TITLE is also omitted, the report name is used.

Note that DBCS characters should not be used in USERDATA, because they are translated when they appear on the screen.

USERID(*name*)

specifies the userid name (1-8 characters) where the report is to be sent to.

The valid characters that can be used in the userid name are as follows:

A through Z
0 through 9
@ \$
**SYSPRT

The first character must not be numeric.

Note: To avoid conflict with some POWER commands, USERID should not be specified as LOCAL or SYSTEM.

Conditions

CCERROR

occurs when an incorrect control character has been specified.

Default action: terminate the task abnormally.

ILLOGIC

RESP2 values:

- 1 Incorrect class
- 3 Incorrect report name
- 4 Incorrect number of copies
- 6 Incorrect priority
- 7 Incorrect forms
- 8 Incorrect destination
- 9 Incorrect node
- 10 Incorrect RSL value
- 13 Incorrect mapset name
- 14 Incorrect map name
- 15 Incorrect USERID
- 16 Incorrect TOKEN
- 18 Incorrect userdata field
- 23 Incorrect PWRSYSID

Default action: terminate the task abnormally.

INVREQ

RESP2 values:

- 22 Unsupported language (RPGII)

Default action: terminate the task abnormally.

IOERR

occurs when an error response is returned from POWER. RESP2 contains the POWER return and feedback codes in the format **rrffvvvv** where:

rr IJBXRETC
ff IJBXREAS
vv offset 4 from IJBXRUSR (PXPRETCD)
vv offset 5 from IJBXRUSR (PXPFBKCD).

IJBXRETC and IJBXREAS are detailed in the DSECT generated by MAPXPCCB.

PXPRETCD and PXPFBKCD are detailed in the DSECT PXPUSER generated as part of PWRSPL TYPE=MAP.

Default action: terminate the task abnormally.

NOSPACE

occurs when the spool files are 90% full. If you continue creating reports without providing more space, IOERR occurs, and you will be unable to close your report.

Default action: terminate the task abnormally.

NOSPOOL

occurs when part, or all, of the POWER interface is unavailable.

RESP2 values:

- 1 No subsystem present
- 2 No report controller interface present

SPOOLOPEN MAPNAME

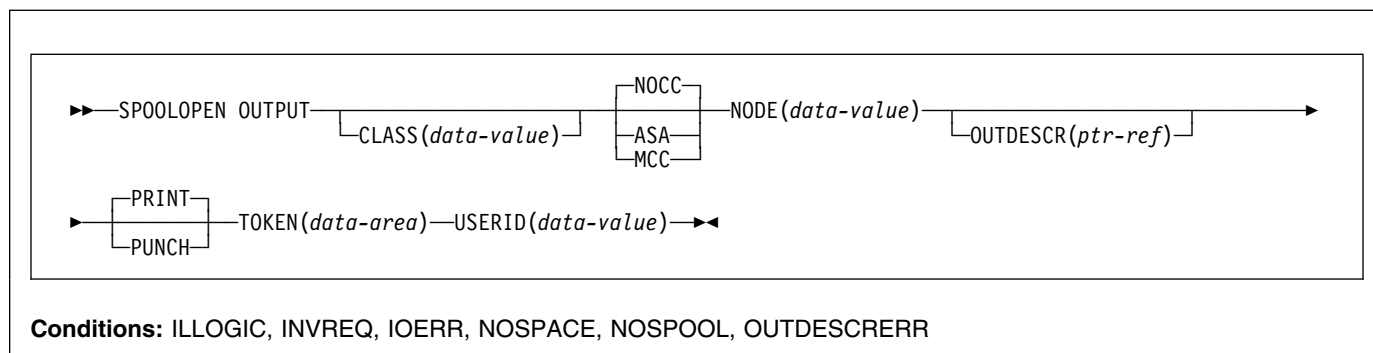
- 6 Interface being terminated
- 7 Interface not active
- 8 Interface being disabled
- 12 Interface has been stopped

Default action: terminate the task abnormally.

RESP2 values:

SPOOLOPEN OUTPUT

Open an output report.



Description

This command opens a spool report for output from CICS to the system spooler and defines its characteristics.

It results in the creation of an output file using the node_id to specify the remote destination and the userid to specify the remote user. This SPOOLOPEN enables users to acquire the token for a report that it expects to create (write). This token is used to identify the report in later SPOOLWRITE and SPOOLCLOSE commands.

When printing on a local device, use the NOCCIASAIMCC parameters to control output formatting. If you do not specify a format, the default value of NOCC is used. NODE(**) and USERID(**) may be used to write the data set directly to the local spool file. For example:

```
EXEC CICS SPOOLOPEN OUTPUT NODE('*')
      USERID('*') CLASS('A') ASA
```

If you do not issue SPOOLCLOSE before the end of the transaction, CICS will perform an implicit SPOOLCLOSE DELETE and write a message to CSMT to alert you to the possible unnecessary retention of resources. You must use the NOHANDLE option, otherwise you get an APST abend.

Options

CLASS(data-value)

is the class designation. This assigns a class to the output data set, and may be used as a selection parameter for the output report. CLASS is a 1-character sender field. CLASS is optional. If it is omitted, the default class specified by the SPOOL system initialization parameter is used.

NOCC or ASA or MCC

NOCC

This specifies that the report about to be created will have no internal formatting controls. When NOCC records are written to the VSE/POWER spool files they are prefixed with hexadecimal zero, X'00'. When the report is printed, the operating system will prefix each record with a carriage-control character that will cause page skipping according to the default operating-system lines-per-page value. This option is not available with the PUNCH option. PRINT must be specified.

ASA

specifies that the report about to be created will have each record prefixed with an ASA carriage-control character, and that this character must be used by the operating system to control formatting when the report is printed. This option is not available with the PUNCH option. PRINT must be specified.

MCC

This specifies that the report about to be created will have each record prefixed with an IBM machine command code carriage-control character, and that this character must be used by the operating system to control formatting when the report is printed. This option is not available with the PUNCH option. PRINT must be specified.

NODE(data-value)

specifies the 8-character identifier of a destination node that the system spooler will use to route the file. It is a sender field.

To write the report directly to the local spool file, use NODE(**) and USERID(**).

SPOOLOPEN OUTPUT

OUTDESCR(ptr-ref)

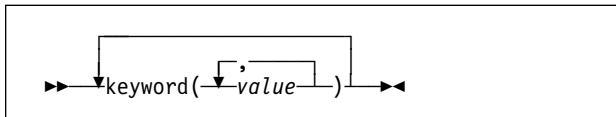
OUTDESCR gives CICS applications the VSE/POWER facility to allow user defined output operands to be specified for batch created reports on the VSE/POWER LST statement. This enables CICS applications to create spool data sets which contain the additional information required by advanced function printing products such as PSF.

Double indirect addressing is used in this command. The ptr-value argument must contain the address of a field that contains the address of a string consisting of a length field and one or more keyword(value) pairs of parameters which are valid user defined output operands specified to VSE/POWER via the DEFINE autostart statement. The *VSE/POWER Administration and Operation* manual gives information on how to use the DEFINE statement.

The user must set up the pointer, the address field, and the string. The format of the string is:

Offset	Length	Contents
0	4	Length (n) of following text string
4	n	Text string of keyword(value) pair(s) separated by spaces.

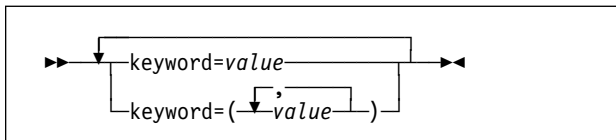
The parameters use the same keywords and values as are used on the VSE/POWER LST statement for user defined output operands but the syntax is slightly different. The syntax of the text string is:



For example:

```
keyword1(value1)
keyword2(value2)
keyword3(value3,value4)
```

This corresponds to the LST syntax:



For example:

```
keyword1=value1
keyword2=value2
keyword3=(value3,value4)
```

CICS returns OUTDESCRERR to the application when there is incorrect or inconsistent use of keywords or values.

The OUTDESCR operand cannot be used to override the NODE, USERID or CLASS operand specifications. The OUTDESCR operand is used to specify additional attributes for the spool data set. For example, to associate a FORMDEF named FORM1 and PAGEDEF

named PAGE1 with a spoolfile for an AFP you could code:

```
EXEC CICS SPOOLOPEN NODE('*') USERID('*')
CLASS('X') OUTDESCR(PARMA)
```

PARMA is a pointer (COBOL, PL/I or C), or register (Assembler). PARMA points to an address field of similar type (address constant in Assembler) that contains the address of a 33-byte area containing the value 29 in bytes 1-4 and the string 'FORMDEF(FORM1) PAGEDEF(PAGE1)' in bytes 5-33.

OUTPUT

indicates that the report is to be written by CICS.

PRINT or PUNCH

PRINT This specifies a listing report, it is the default setting.

PUNCH This specifies a card image report.

TOKEN(data-area)

This is the CICS-allocated token used to identify a report. It is an 8-byte value aligned on a fullword boundary. It is a receiver on SPOOLOPEN and a sender on all other commands.

USERID(data-value)

Userid is the VSE/POWER identifier of the program or user that will process the report, and must be from 1–8 characters and must not include blank (X'40') or null (X'00') characters. The report carries this identifier and it is used to select the report at its destination. It is a sender field. For an output file, the resource security level (RSL) of the report must match the OPERRSL key of the terminal at which the transaction is running.

Validity checking is performed for USERID. Checks are made for blanks (X'40'), and nulls (X'00').

To write the report directly to the local spool file, code USERID('*') and NODE('*').

Conditions

Note: RESP2 values of 256 and over (unless listed below) are internal, except for IOERR and OUTDESCRERR. If a RESP2 value greater than 256 is returned with any RESP value except IOERR or OUTDESCRERR, contact your IBM support center.

ILLOGIC

RESP2 values:

- 1 Invalid class specified
- 2 Invalid PUNCH report format specified

Default action: terminate the task abnormally.

INVREQ

RESP2 values:

- 1 Unsupported language (RPG II).

- 24 Invalid OUTDESCR address or invalid address in area. Unsupported language (RPG II).
- 322 Invalid USERID parameter.
- 333 Invalid NODE parameter.

Default action: terminate the task abnormally.

IOERR

Indicates an input/output error. The RESP2 codes show the decimal representation of the hexadecimal POWER return and feedback codes documented in the PWRSPPL copybook. For example, a RESP2 value of 2057 is equivalent to a POWER return code of X'0809', which means "INVALID USER/REMOTE ID." of a currently open spool file.

Default action: terminate the task abnormally.

NOSPACE

Indicates that there is no space on the POWER spool file to add records. If you do not have the report controller installed, spool files that are open for input are deleted by VSE/POWER. If you have the report controller installed, reports that are open for input with the LOGICAL recovery option are deleted by VSE/POWER, and reports that are open for input with the PHYSICAL recovery option are kept.

Default action: terminate the task abnormally.

NOSPOOL

RESP2 values:

- 1 The VSE/POWER interface is not available. No subsystem is present.
- 2 The VSE/POWER interface is not available. No report controller interface is present.

Default action: terminate the task abnormally.

OUTDESCRERR

Indicates an error in the string passed with the OUTDESCR parameter. RESP2 contains POWER return codes as documented in the PWRSPPL copy book.

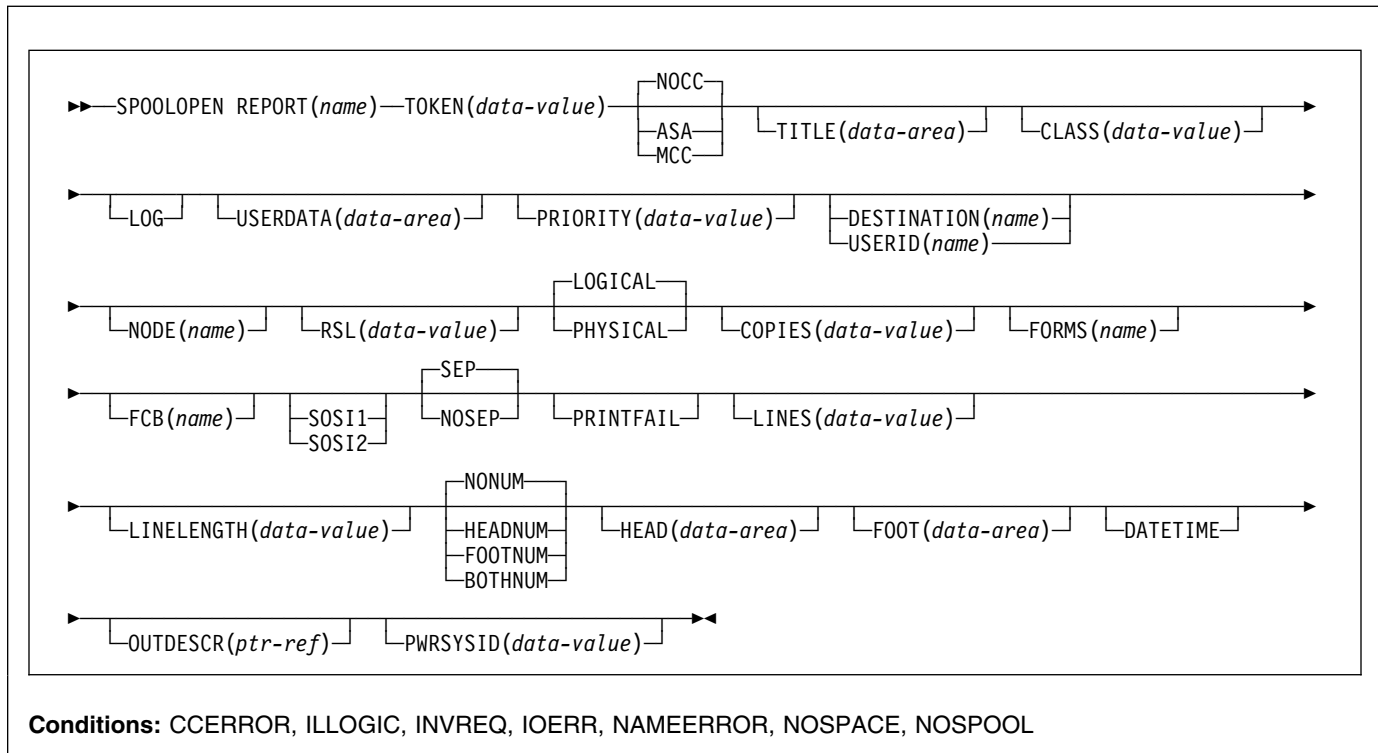
RESP2 values:

- X'083F'(ptr-ref) no keyword passed.
- X'0840'(ptr-ref) the keyword passed has no corresponding DEFINE keyword.
- X'0841'(ptr-ref) the value passed for a keyword does not match the DEFINE specification.

Default action: terminate the task abnormally.

SPOOLOPEN REPORT (ASA, MCC or NOCC)

Open ASA, MCC, or NOCC report.



Description

SPOOLOPEN REPORT is a function of the report controller facility. For more information about the report controller, see the *CICS Report Controller User's Guide* and the *CICS Report Controller Planning Guide*.

Options

ASA

specifies that the report is created using an ASA control character at the beginning of each line of data, as follows:

- blank** Space 1 line before writing the data.
- 0** Space 2 lines before writing the data.
- Space 3 lines before writing the data.
- +** Suppress space (that is, print on the same line as the previous line).
- 1** Skip to line 1 of new page before writing the data.

You should ensure that valid carriage control characters are used. If not, the CCERROR condition occurs.

If you specify LINES(0), you can skip to channels 2 through 12. LINES(0) means that no automatic page breaks are inserted. If you want automatic page breaks, you cannot skip to channels 2 through 12.

ASA data is always converted to MCC format. You can improve performance by writing several ASA lines in one SPOOLWRITE REPORT command. The resulting increase in speed may, however, be offset by the increased storage needed for buffers.

BOTHNUM

specifies that page numbers are to appear on both headings and footings in the printed report. The page number appears as the last 6 characters of each of these lines with leading zeros suppressed. If you do not specify HEAD or FOOT, the rest of the heading or footing is blank.

CLASS(data-value)

specifies the POWER output class (1 character A–Z, or numeral 0–9) for reports in all formats except JCL. The default class is specified by the SPOOL system initialization parameter.

COPIES(*data-value*)

specifies the number of copies (signed halfword binary 1–255) requested for this report. If not specified, 1 is assumed.

DATETIME

specifies that the date and time the report was created, appear in the heading of each page.

The length of the DATETIME field is always 14 characters. The format of the date is determined by the system programmer at CICS generation time. The format of the time is always HH:MM.

The possibilities are:

DD/MM/YY HH:MM
MM/DD/YY HH:MM
YY/MM/DD HH:MM

DESTINATION(*name*)

specifies the destination name (1-8 characters) at which the report is to be printed.

The valid characters that can be used in the destination name are as follows:

A through Z
0 through 9
@ \$ #
**SYSPRT

The first character must not be numeric.

Note: To avoid conflict with some POWER commands, DESTINATION should not be specified as LOCAL or SYSTEM.

FCB(*name*)

specifies the name of the forms control buffer (FCB) to be used for the report.

FOOT(*data-area*)

specifies the field (1–205 characters) that contains the footing to be used in the report. CICS does not check this length so care must be taken that it is the same as that specified by the LINELENGTH option (or its default), otherwise, the results are not predictable.

If the report format is ASA or MCC, the first character of the field is used for the carriage control character.

See Table 17 on page 314 for the right-hand characters of when using FOOTNUM.

FOOTNUM

specifies that a page number is to appear as the last 6 characters of the footing of the report, the first of which is a separator space. If no footing is supplied in SPOOLOPEN REPORT, a blank footing line is inserted in the report.

FORMS(*name*)

specifies the name of the stationery (1-4 characters) to be used to print the report. At print time, a match is

made between the forms specified for the report and the forms type for the printer. (The forms type for a printer can be specified or altered using the CEMS and CEOS transactions.) If they match, the report is printed; otherwise, the report remains “in use” until the correct forms type is selected.

Note: It must be stressed that no physical check is made regarding the size of the paper in the printer; if the forms type is right, the report is printed, regardless of what paper is actually loaded.

The valid characters that can be used in the forms name are as follows:

A through Z
0 through 9
- . / @ \$ #

HEAD(*data-area*)

specifies the field (1–205 characters) that contains the heading to be used in the report. CICS does not check this length so care must be taken that it is the same as that specified by the LINELENGTH option (or its default), otherwise, the results are not predictable.

See Table 17 on page 314 for the right-hand characters of HEAD when using DATETIME and HEADNUM.

If the report format is ASA or MCC, the first character of the field is used for the carriage control character.

HEADNUM

specifies that a page number is to appear as the last 6 characters of the heading of the report. If no heading is supplied at SPOOLOPEN REPORT, a blank heading line is inserted in the report.

LINELENGTH(*data-value*)

specifies the length (signed halfword binary) of each line of the report. This value is specified in SPOOLOPEN REPORT and remains fixed for the entire life of the report. The valid range is 1–205 characters.

If HEADNUM, FOOTNUM, or BOTHNUM is specified, LINELENGTH must be at least 6 for NOCC type reports and 7 for ASA or MCC type reports.

If DATETIME is specified as well, LINELENGTH must be at least 20 for NOCC type reports and 21 for ASA or MCC type reports.

You are recommended to set LINELENGTH to no more than the width of the printer carriage minus 1. If LINELENGTH exceeds the width of the printer carriage, the format of the printed report may be unpredictable.

For ASA and MCC reports, LINELENGTH should include the carriage control character.

If this option is omitted, the following apply:

- A JCL report is always forced to 80 characters regardless of the LINELENGTH option.
- The LINELENGTH option is not meaningful for reports in ESC, MAP, SCS (with NOCONV), or T3270 (with NOCONV) formats.

SPOOLOPEN REPORT (ASA, MCC or NOCC)

- For ASA or MCC format report, 133 is the default.
- For NOCC, SCS, or T3270 formats, 132 is the default.

LINES(*data-value*)

specifies the number of lines (signed halfword binary) accepted in each page of the report.

When writing to the report, if this number is exceeded without a “skip to new page” being encountered, a “default page break” takes place.

The valid range is 0 through 99.

When the LINES option is not specified the system default is used.

If LINES(0) is explicitly specified, no page formatting is attempted and headings or footings are not allowed.

LOG

specifies that a log report is to be opened. A log report is defined as a report that can be written to by a number of applications concurrently. A log report stays open until it is closed explicitly, or it is closed implicitly at CICS termination.

A log report can be in any format.

LOGICAL

specifies that the report has logical recovery.

Reports are not committed until a POWER checkpoint is issued. With LOGICAL recovery, a POWER checkpoint is issued only when the report is closed, either by a SPOOLCLOSE REPORT command, or when a syncpoint is issued.

If an abnormal termination occurs while a task is active, the report is backed out to the last POWER checkpoint.

Logical recovery is not supported for LOG reports.

MCC

specifies that the report is created using machine carriage control characters, as follows:

Hex	Result
------------	---------------

0B	Space 1 line immediately.
13	Space 2 lines immediately.
1B	Space 3 lines immediately.
09	Space 1 line after write.
11	Space 2 lines after write.
19	Space 3 lines after write.
89	Skip to channel 1 after write.
8B	Skip to channel 1 immediately.
01	Write without spacing.
03	No operation.

It is the responsibility of the application to ensure that the first character of each line to print is a valid carriage control character. If an incorrect carriage control character is encountered, the CCERROR condition occurs.

NOCC

specifies that no carriage control is used when creating this report. You control page breaks using the LINES option. A new page occurs whenever the LINES option value is exceeded.

NODE(*name*)

specifies the (1-8 characters) name of the target node, to which the file is to be routed by the system spooler (POWER). If you specify NODE and USERID, the file may be routed to a CICS printer. If you specify NODE alone, the file may be routed only to a system printer.

The valid characters that can be used in the node name are as follows:

A through Z
0 through 9
@ \$ #

The first character must not be numeric.

NONUM

specifies that no automated page numbering is to take place.

NOSEP

specifies that no separator pages are to appear with the printed report.

OUTDESCR(*ptr-ref*)

OUTDESCR gives CICS applications the VSE/POWER facility to allow user defined output operands to be specified for batch created reports on the VSE/POWER LST statement. This enables CICS applications to create reports which contain the additional information required by advanced function printing products such as PSF.

Double indirect addressing is used in this command. The ptr-ref argument must contain the address of a field that contains the address of a string consisting of a length field and one or more keyword(value) pairs of parameters which are valid user defined output operands specified to VSE/POWER via the DEFINE autostart statement. The *VSE/POWER Administration and Operation* manual gives information on how to use the DEFINE statement.

PHYSICAL

specifies that this report is to have physical recovery. When creating a report with physical recovery each SPOOLWRITE REPORT command, when completed, results in a physical write to disk of all data supplied in the command (with a POWER checkpoint). Physical recovery makes heavy use of input/output and should be carefully considered before being selected.

PRINTFAIL

specifies that if CEPW abends (due to a CICS abend) before a report that is being printed has been closed, when CEPW recovers, it does not attempt to reprint this report. The report status is set to ERRPRT. Operator

intervention is required before printing of the report can continue.

PRIORITY(*data-value*)

specifies the priority (signed halfword binary) of the report. Reports destined for the same destination that are READY are printed in order of priority. When specified, the value must be in the range from 1 through 9 (where 9 is highest priority). If not specified, a value of 3 is assumed.

PWRSYSID(*data-value*)

specifies the 1-character VSE/POWER SYSID to be associated with this report. When specified, the value must be in the range 1 through 9. If not specified, the report is not assigned a VSE/POWER SYSID value.

REPORT(*name*)

specifies the name (2–8 characters) of the report. This option is required for every command.

An attempt to open a log report with the same name as one already open is rejected.

The valid characters that can be used in the report name are as follows:

A through Z
0 through 9
- . / @ \$ #

The first character must not be numeric.

Do **not** use any of the following names, because you will not be able to manipulate them using CEMS or CEOS, or using POWER operator commands:

ALL
DEV
FREE
HOLD
LOCAL
PNET
RJE
STATUS.

RSL(*data-value*)

specifies the security key (1–6 characters) that is associated with the report. The value must be in the range 1–24 or "PUBLIC" (127). If the RSL option is not specified, PUBLIC is assumed.

When the report is created, only terminal operators with the same resource level specification are able to alter its characteristics. The report can only be printed on printers with the same RSL value specified in OPERRSL.

The security key is checked at SPOOLCLOSE REPORT of a continuous report and at SPOOLOPEN RESUME. If the terminal operator cannot match the security key, the NOTAUTH condition occurs.

After JCL reports are closed, RSL is set to 1.

SEP

specifies that two separator pages should be printed preceding the report, and two following.

SOSI1

specifies that shift-out/shift-in control sequences take a print position for a report containing double-byte character set (DBCS) fields. Each control sequence prints as a blank. SOSI1 and SOSI2 are mutually exclusive. There is no default specification; if you do not specify either SOSI1 or SOSI2, no indication of a SOSI attribute is made at the time the report is created.

SOSI2

specifies that shift-out/shift-in control sequences do not take a print position for a report containing double-byte character set (DBCS) fields. SOSI1 and SOSI2 are mutually exclusive. There is no default specification; if you do not specify either SOSI1 or SOSI2, no indication of a SOSI attribute is made at the time the report is created.

TITLE(*data-area*)

specifies the title to be displayed on the CEMS or CEOS report characteristics panel. The title is a 32-character field which is not subject to any restrictions or validation. The default title is the REPORT name. USERDATA defaults to the first 16 characters of TITLE.

Note that DBCS characters can be used in TITLE, but if you do not code USERDATA, and let it default to the first 16 characters of TITLE, DBCS characters are translated when they appear on the screen.

TOKEN(*data-value*)

a token is used by CICS to identify the report during creation. The option is required on all commands.

You must specify an 8-character area (that must be word-aligned) and store the value returned by CICS after a successful SPOOLOPEN REPORT.

USERDATA(*data-area*)

specifies a report description (1-16 characters) or other reference information to be displayed on the CEMS or CEOS report list panel. If this option is omitted, the first 16 printable characters of the title are used. If TITLE is also omitted, the report name is used.

Note that DBCS characters should not be used in USERDATA, because they are translated when they appear on the screen.

USERID(*name*)

specifies the userid name (1-8 characters) where the report is to be sent to.

The valid characters that can be used in the userid name are as follows:

A through Z
0 through 9
@ \$
**SYSPRT

SPOOL OPEN REPORT (ASA, MCC or NOCC)

The first character must not be numeric.

Note: To avoid conflict with some POWER commands, USERID should not be specified as LOCAL or SYSTEM.

HEAD (2 lines)	DATETIME (14 chars)	HEADNUM (6 chars)
LINELENGTH()		
FOOT (2 lines)		FOOTNUM (6 chars)

HEAD (2 lines)	DATETIME (14 chars)	HEADNUM (7 chars)
LINELENGTH()		
FOOT (2 lines)		FOOTNUM (7 chars)

Conditions

CCERROR

occurs when an invalid control character is specified.

ILLOGIC

RESP2 values:

- 1 Incorrect class
- 3 Incorrect report name
- 4 Incorrect number of copies
- 6 Incorrect priority
- 7 Incorrect forms
- 8 Incorrect destination
- 9 Incorrect node
- 10 Incorrect RSL value
- 11 Incorrect lines per page
- 12 Incorrect line length
- 15 Incorrect USERID
- 16 Incorrect TOKEN
- 18 Incorrect userdata field
- 19 Incorrect LINELENGTH for DATETIME, HEADER, FOOTNUM, or BOTHNUM
- 23 Incorrect PWRSYSID

Default action: terminate the task abnormally.

INVREQ

RESP2 values:

- 22 Unsupported language (RPGII)
- 24 Invalid OUTDESCR address or invalid address in area.

Default action: terminate the task abnormally.

IOERR

occurs when an error response is returned from POWER. RESP2 contains the POWER return and feedback codes in the format **rrffvvvv** where:

rr IJBXRETC.

ff IJBXREAS.

vv offset 4 from IJBXRUSR (PXPRETCD).

vv offset 5 from IJBXRUSR (PXPFBKCD).

IJBXRETC and IJBXREAS are detailed in the DSECT generated by MAPXPCCB.

PXPRETCD and PXPFBKCD are detailed in the DSECT PXPUSER generated as part of PWRSPY TYPE=MAP.

Default action: terminate the task abnormally.

NAMEERROR

occurs when an application tries to open a LOG report that already exists.

NOSPACE

occurs when the spool files are 90% full. If you continue creating reports without providing more space, IOERR occurs, and you will be unable to close your report.

Default action: terminate the task abnormally.

NOSPOOL

occurs when part, or all, of the POWER interface is unavailable.

RESP2 values:

- 1 No subsystem present
- 2 No report controller interface present
- 6 Interface being terminated
- 7 Interface not active
- 8 Interface being disabled
- 12 Interface has been stopped

Default action: terminate the task abnormally.

OUTDESCRERR

Indicates an error in the input string that is detected by VSE/POWER. RESP2 contains the VSE/POWER return codes that are documented in the PWRSPY copy book.

RESP2 values:

X'083F' (**ptr-ref**) no keyword passed.

X'0840' (**ptr-ref**) the keyword passed has no corresponding DEFINE keyword.

X'0841' (**ptr-ref**) the value passed for a keyword does not match the DEFINE specification.

Default action: terminate the task abnormally.

SPOOLOPEN REPORT (JCL)

Open JCL format report.

```
▶▶ SPOOLOPEN REPORT(name)—JCL—TOKEN(data-value)—LOG▶▶
```

Conditions: CCERROR, ILLOGIC, INVREQ, IOERR, NOSPACE, NOSPOOL

Description

SPOOLOPEN REPORT (JCL) is a function of the report controller facility. For more information about the report controller, see the *CICS Report Controller User's Guide* and the *CICS Report Controller Planning Guide*.

Options

JCL

specifies that the report is to be in JCL format, and that it is to be written to the POWER RDR queue as input to a batch program. It is the responsibility of the application to ensure correct JCL syntax of the report. A JCL report has a forced LINELENGTH of 80 characters.

If a report submitted to POWER does not contain POWER JECL, the required JECL is inserted by POWER, and the characteristics of the JCL report default to those allocated by POWER.

The only other options you can specify with JCL are LOGICAL or LOG, but you cannot specify both for the same JCL report.

LOG

specifies that a log report is to be opened. A log report is defined as a report that can be written to by a number of applications concurrently. A log report stays open until it is closed explicitly, or it is closed implicitly at CICS termination.

A log report can be in any format.

REPORT(*name*)

specifies the name (2–8 characters) of the report. This option is required for every command.

An attempt to open a log report with the same name as one already open is rejected.

The valid characters that can be used in the report name are as follows:

```
A through Z
0 through 9
- . / @ $ #
```

The first character must not be numeric.

Do **not** use any of the following names, because you will not be able to manipulate them using CEMS or CEOS, or using POWER operator commands:

```
ALL
DEV
FREE
HOLD
LOCAL
PNET
RJE
STATUS
```

TOKEN(*data-value*)

used by CICS to identify the report during creation. The option is required on all commands.

You must specify an 8-character area (that must be word aligned) and store the value returned by CICS after a successful SPOOLOPEN REPORT.

Conditions

CCERROR

occurs when an incorrect control character has been specified.

Default action: terminate the task abnormally.

ILLOGIC

RESP2 values:

3 Incorrect report name

16 Incorrect TOKEN

Default action: terminate the task abnormally.

INVREQ

RESP2 values:

22 Unsupported language (RPGII)

Default action: terminate the task abnormally.

SPOOL OPEN REPORT (JCL)

IOERR

occurs when an error response is returned from POWER. RESP2 contains the POWER return and feedback codes in the format **rrffvvvv** where:

rr IJBXRETC.

ff IJBXREAS.

vv offset 4 from IJBXRUSR (PXPRETCD).

vv offset 5 from IJBXRUSR (PXPFBKCD).

IJBXRETC and IJBXREAS are detailed in the DSECT generated by MAPXPCCB.

PXPRETCD and PXPFBKCD are detailed in the DSECT PXPUSER generated as part of PWRSPD TYPE=MAP.

Default action: terminate the task abnormally.

NOSPACE

occurs when the spool files are 90% full. If you continue creating reports without providing more space, IOERR occurs, and you will be unable to close your report.

Default action: terminate the task abnormally.

NOSPOOL

occurs when part, or all, of the POWER interface is unavailable.

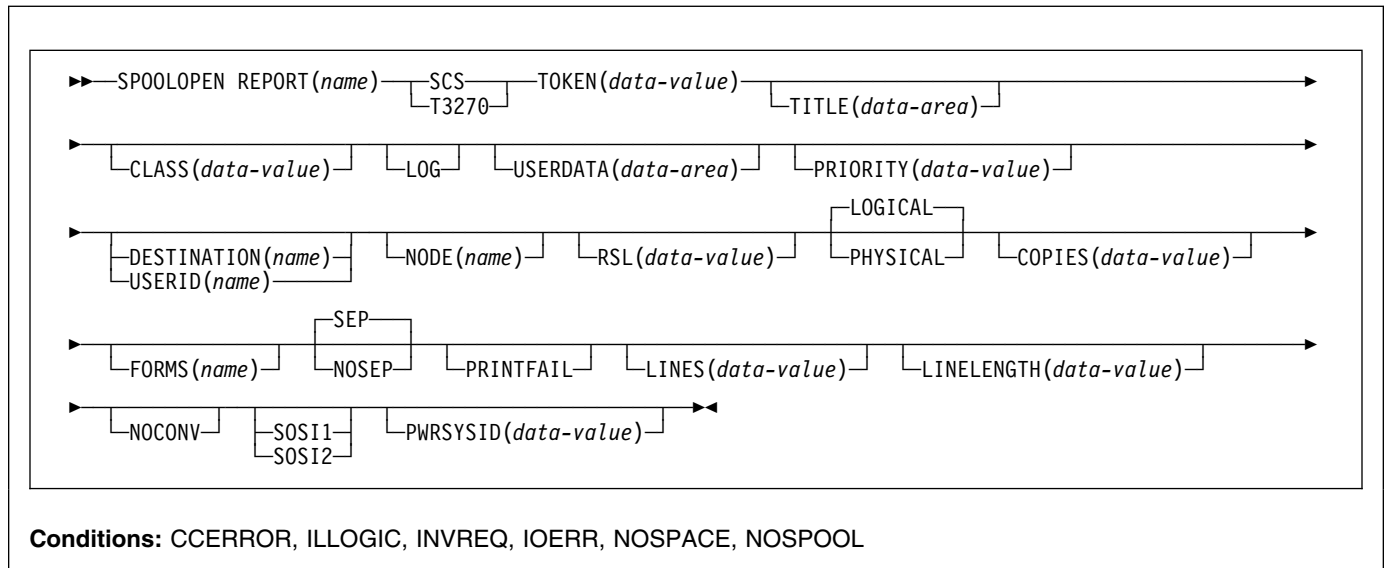
RESP2 values:

- 1** No subsystem present
- 2** No report controller interface present
- 6** Interface being terminated
- 7** Interface not active
- 8** Interface being disabled
- 12** Interface has been stopped

Default action: terminate the task abnormally.

SPOOLOPEN REPORT (SCS/T3270)

Open SCS or T3270 report.



Description

SPOOLOPEN REPORT (SCS/T3270) is a function of the report controller facility. For more information about the report controller, see the *CICS Report Controller User's Guide* and the *CICS Report Controller Planning Guide*.

Options

CLASS(data-value)

specifies the POWER output class (1 character A–Z, or numeral 0–9) for reports in all formats except JCL. The default class is specified by the SPOOL system initialization parameter.

COPIES(data-value)

specifies the number of copies (signed binary halfword 1–255) requested for this report. If not specified, 1 is assumed.

DESTINATION(name)

specifies the destination name (1-8 characters) at which the report is to be printed.

The valid characters that can be used in the destination name are as follows:

A through Z
0 through 9
@ \$ #
**SYSPRT

The first character must not be numeric.

Note: To avoid conflict with some POWER commands, DESTINATION should not be specified as LOCAL or SYSTEM.

FORMS(name)

specifies the name of the stationery (1-8 characters) to be used to print the report. At print time, a match is made between the forms specified for the report and the forms type for the printer. (The forms type for a printer can be specified or altered using the CEMS and CEOS transactions). If they match, the report is printed, otherwise the report remains "in use" until the correct forms type is selected.

Note: It must be stressed that no physical check is made regarding the size of the paper in the printer; if the forms type is right, the report is printed, regardless of what paper is actually loaded.

The valid characters that can be used in the forms name are as follows:

A through Z
0 through 9
- . / @ \$ #

LINELENGTH(data-value)

(cannot be specified for SCS, with NOCONV, and T3270, with NOCONV). Specifies the length of each line (signed binary halfword) of the report. This value is specified at SPOOLOPEN REPORT and remains fixed for the entire life of the report. The valid range is 1–205 characters.

If DATETIME is specified as well, LINELENGTH must be greater than or equal to 20.

SPOOLOPEN REPORT (SCS/T3270)

If HEADNUM, FOOTNUM, or BOTHNUM is specified as well, LINELENGTH must be greater than or equal to 6.

When deciding the LINELENGTH for a report, you should consider the type of printer on which the report is printed. You are recommended to set the LINELENGTH to no more than the width of the printer carriage minus 1. If the LINELENGTH exceeds the width of the printer carriage, the format of the printed report may be unpredictable.

For ASA and MCC reports, LINELENGTH should include the carriage control character.

If this option is omitted, the following applies:

- A JCL report is always forced to 80 characters regardless of the LINELENGTH option.
- The LINELENGTH option is not meaningful for reports in ESC, MAP, SCS (with NOCONV), or T3270 (with NOCONV) formats.
- For ASA or MCC format report, 133 is the default.
- For NOCC, SCS, or T3270 formats, 132 is the default.

LINES(*data-value*)

specifies, as a signed binary halfword, for SCS (without NOCONV) and T3270 (without NOCONV) format reports, the number of lines accepted in each page of the report. (Cannot be specified for SCS, with NOCONV, and T3270, with NOCONV.)

When writing to the report, if this number is exceeded without a “skip to new page” being encountered, a “default page break” takes place.

The valid range is 0 through 99.

When the LINES option is not specified the system default is used.

If LINES(0) is explicitly specified, no page formatting is attempted and headings or footings are not allowed.

LOG

specifies that a log report is to be opened. A log report is defined as a report that can be written to by a number of applications concurrently. A log report stays open until it is closed explicitly, or it is closed implicitly at CICS termination.

A log report can be in any format.

LOGICAL

specifies that the report has logical recovery.

Reports are not committed until a POWER checkpoint is issued. With LOGICAL recovery, a POWER checkpoint is issued only when the report is closed, either by a SPOOLCLOSE REPORT command, or when a syncpoint is issued.

If an abnormal termination occurs while a task is active, the report is backed out to the last POWER checkpoint.

Logical recovery is not supported for LOG reports.

NOCONV

specifies that no conversion to MCC format is required. If specified, the report can only be printed on an appropriate CICS printer. This option is applicable only to SCS and T3270 formats, and allows the user complete control over the data sent to the printer. However, if incorrect data is sent, the writer task may abend.

When printing reports with this option, all the data specified with each SPOOLWRITE REPORT is sent to the printer. The printer buffer is then printed before the data associated with the next SPOOLWRITE REPORT is processed.

NODE(*name*)

specifies the (1-8 characters) name of the target node, to which the file is to be routed by the system spooler (POWER). If you specify NODE and USERID, the file may be routed to a CICS printer. If you specify NODE alone, the file may be routed only to a system printer.

The valid characters that can be used in the node name are as follows:

A through Z
0 through 9
@ \$ #

The first character must not be numeric.

NOSEP

specifies that no separator pages are to appear with the printed report.

PHYSICAL

specifies that this report is to have physical recovery. When creating a report with physical recovery each SPOOLWRITE REPORT command, when completed, results in a physical write to disk of all data supplied in the command (with a POWER checkpoint). Physical recovery makes heavy use of input/output, and should be carefully considered before being selected.

PRINTFAIL

specifies that should CEPW abend (due to a CICS abend, and so on) before a report that is being printed has been closed, then when CEPW has recovered, it does not attempt to reprint this report. The report status is set to ERRPRT. Operator intervention is required before printing of the report can continue.

PRIORITY(*data-value*)

specifies the priority (signed binary halfword) of the report. Reports destined for the same destination that are READY are printed in order of priority. When specified, the value must be in the range from 1 through 9 (where 9 is highest priority). If not specified, a value of 3 is assumed.

PWRSYSID(*data-value*)

specifies the 1-character VSE/POWER SYSID to be associated with this report. When specified, the value

must be in the range 1 through 9. If not specified, the report is not assigned a VSE/POWER SYSID value.

REPORT(*name*)

specifies the name (2–8 characters) of the report. This option is required for every command.

An attempt to open a log report with the same name as one already open is rejected.

The valid characters that can be used in the report name are as follows:

A through Z
0 through 9
- . / @ \$ #

The first character must not be numeric.

Do **not** use any of the following names, because you will not be able to manipulate them using CEMS or CEOS, or using POWER operator commands:

ALL
DEV
FREE
HOLD
LOCAL
PNET
RJE
STATUS.

RSL(*data-value*)

specifies the security key (1–6 characters) that is associated with the report. The value must be in the range of 1 through 24 or "PUBLIC" (127). If the RSL option is not specified, public is assumed.

Once the report is created, only terminal operators with the same resource level specification (specified in the sign-on table) are able to alter its characteristics. The report can only be printed on printers with the same RSL value specified in OPERRSL.

The security key is checked at SPOOLCLOSE REPORT of a continuous report and at SPOOLOPEN RESUME. If the terminal operator cannot match the security key, the NOTAUTH condition occurs.

After JCL reports are closed, RSL is set to 1.

SCS

specifies that the report being opened uses the SNA character string. It is the responsibility of the application programmer to ensure correct carriage control characters.

For SCS, a SPOOLWRITE REPORT is formatted into a whole number of MCC lines. When an SCS (with NOCONV) report is printed, the destination is checked for SCS support. If no SCS support is specified for the destination, message DFHRC5459 is sent to CSPW.

SEP

specifies that two separator pages should be printed preceding the report, and two following.

SOSI1

specifies that shift-out/shift-in control sequences takes a print position for a report containing double-byte character set (DBCS) fields. Each control sequence prints as a blank. SOSI1 and SOSI2 are mutually exclusive. There is no default specification; if you do not specify either SOSI1 or SOSI2, no indication of a SOSI attribute is made at the time the report is created.

SOSI2

specifies that shift-out/shift-in control sequences does not take a print position for a report containing double-byte character set (DBCS) fields. SOSI1 and SOSI2 are mutually exclusive. There is no default specification; if you do not specify either SOSI1 or SOSI2, no indication of a SOSI attribute is made at the time the report is created.

TITLE(*data-area*)

specifies the title to be displayed on the CEMS/CEOS report characteristics panel. The title is a 32-character field that is not subject to any restrictions or validation. The default title is the REPORT name. USERDATA defaults to the first 16 characters of TITLE.

Note that DBCS characters can be used in TITLE, but if you do not code USERDATA, and let it default to the first 16 characters of TITLE, DBCS characters are translated when they appear on the screen.

TOKEN(*data-value*)

used by CICS to identify the report during creation. The option is required on all commands.

You must specify an 8-character area (that must be word-aligned) and store the value returned by CICS after a successful SPOOLOPEN REPORT.

T3270

specifies that the report to be created is in terminal 3270 format. A report written in T3270 format can contain any control characters normally supported by the 3270-type printers. It is the responsibility of the application to ensure that the control characters being used are valid because the report controller attempts no validation.

For T3270, SPOOLWRITE REPORT is formatted into a whole number of MCC lines. When a T3270 (with NOCONV) report is printed, the destination is checked for 3270 support. If no 3270 support is available, message DFHRC5459 is sent to CSPW.

USERDATA(*data-area*)

specifies a report description (1-16 characters) or other reference information to be displayed on the CEMS/CEOS report list panel. If this option is omitted, the first 16 printable characters of the title are used. If TITLE is also omitted, the report name is used.

Note that DBCS characters should not be used in USERDATA, because they are translated when they appear on the screen.

SPOOLOPEN REPORT (SCS/T3270)

USERID(*name*)

specifies the userid name (1-8 characters) where the report is to be sent to.

The valid characters that can be used in the userid name are as follows:

A through Z
0 through 9
@ \$
**SYSPRT

The first character must not be numeric.

Note: To avoid conflict with some POWER commands, USERID should not be specified as LOCAL or SYSTEM.

vv offset 4 from IJBXRUSR (PXPRETCD)
vv offset 5 from IJBXRUSR (PXPFBKCD).

IJBXRETC and IJBXREAS are detailed in the DSECT generated by MAPXPCCB.

PXPRETCD and PXPFBKCD are detailed in the DSECT PXPUSER generated as part of PWRSPPL TYPE=MAP.

Default action: terminate the task abnormally.

NOSPACE

occurs when the spool files are 90% full. If you continue creating reports without providing more space, IOERR occurs, and you will be unable to close your report.

Default action: terminate the task abnormally.

Conditions

CCERROR

occurs when an incorrect control character has been specified.

Default action: terminate the task abnormally.

ILLOGIC

RESP2 values:

- 1 Incorrect class
- 3 Incorrect report name
- 4 Incorrect number of copies
- 6 Incorrect priority
- 7 Incorrect forms
- 8 Incorrect destination
- 9 Incorrect node
- 10 Incorrect RSL value
- 11 Incorrect lines per page
- 12 Incorrect line length
- 15 Incorrect USERID
- 16 Incorrect TOKEN
- 18 Incorrect userdata field
- 23 Incorrect PWRSYSID

Default action: terminate the task abnormally.

INVREQ

RESP2 values:

- 22 Unsupported language (RPGII)

Default action: terminate the task abnormally.

Default action: terminate the task abnormally.

IOERR

occurs when an error response is returned from POWER. RESP2 contains the POWER return and feedback codes in the format **rrffvvvv** where:

rr IJBXRETC
ff IJBXREAS

NOSPOOL

RESP2 values:

- 1 No subsystem present
- 2 No report controller interface present
- 6 Interface being terminated
- 7 Interface not active
- 8 Interface being disabled
- 12 Interface has been stopped

Default action: terminate the task abnormally.

SPOOLOPEN RESUME

Reopen a report.

```
►►—SPOOLOPEN RESUME REPORT(name)—TOKEN(data-value)—◄◄
```

Conditions: ILLOGIC, IOERR, INVREQ, NAMEERROR, NOSPACE, NOSPOOL, NOTAUTH, WRONGSTAT

Description

SPOOLOPEN RESUME is a function of the report controller facility. For more information about the report controller, see the *CICS Report Controller User's Guide* and the *CICS Report Controller Planning Guide*.

Options

REPORT(*name*)

specifies the name (2–8 characters) of the report. This option is required for every command.

An attempt to open a log report with the same name as one already open is rejected.

The valid characters that can be used in the report name are as follows:

```
A through Z
0 through 9
- . / @ $ #
```

The first character must not be numeric.

Do **not** use any of the following names, because you will not be able to manipulate them using CEMS or CEOS, or using POWER operator commands:

```
ALL
DEV
FREE
HOLD
LOCAL
PNET
RJE
STATUS.
```

RESUME

when used in a SPOOLOPEN REPORT command, the report controller requires the report being opened to have previously been closed with the RESUME option, that is, the status of the report must be RESUME. This option cannot be used for JCL reports or for reports that have NODE specified on the associated SPOOLOPEN REPORT command.

You should issue a SPOOLCLOSE RESUME command before you issue a SYNCPOINT, and a SPOOLOPEN

RESUME after the SYNCPOINT. Otherwise, the SYNCPOINT results in the report being closed in READY status. However, there could be a transaction failure after the report has been closed, and before the SYNCPOINT. This could result in, for instance, file updates being backed out, while the report of file updates is not backed out.

TOKEN(*data-value*)

used by CICS to identify the report during creation. The option is required on all commands.

You must specify an 8-character area (that must be word aligned) and store the value returned by CICS after a successful SPOOLOPEN.

Conditions

ILLOGIC

RESP2 values:

- 3** Incorrect report name
- 16** Incorrect TOKEN

Default action: terminate the task abnormally.

INVREQ

RESP2 values:

- 22** Unsupported language (RPGII)

Default action: terminate the task abnormally.

IOERR

occurs when an error response is returned from POWER. RESP2 contains the POWER return and feedback codes in the format **rrffvvvv** where:

```
rr IJBXRETc
ff IJBXREAS
vv offset 4 from IJBXRUSR (PXPRETCD)
vv offset 5 from IJBXRUSR (PXPFBKCD).
```

IJBXRETc and IJBXREAS are detailed in the DSECT generated by MAPXPCCB.

PXPRETCD and PXPFBKCD are detailed in the DSECT PXPUSER generated as part of PWRSPPL TYPE=MAP.

Default action: terminate the task abnormally.

SPOOLOPEN RESUME

NAMEERROR

occurs when an application has specified the wrong report name. when issuing a command

Default action: terminate the task abnormally.

NOSPACE

occurs when the spool files are 90% full. If you continue creating reports without providing more space, IOERR occurs, and you will be unable to close your report.

Default action: terminate the task abnormally.

NOSPOOL

RESP2 values:

- 1 No subsystem present
- 2 No report controller interface present
- 6 Interface being terminated
- 7 Interface not active
- 8 Interface being disabled
- 12 Interface has been stopped

Default action: terminate the task abnormally.

NOTAUTH

occurs when a resource security check has failed. The reasons for the failure are the same as for abend code AEY7, as listed in the *VSE/ESA Messages and Codes Volume 3* manual.

Default action: terminate the task abnormally.

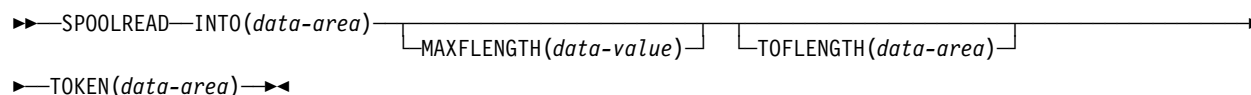
WRONGSTAT

occurs when a SPOOLOPEN RESUME command is issued but the report has not previously been closed as resumable.

Default action: terminate the task abnormally.

SPOOLREAD

Read a spool record.



Conditions: ENDFILE, INVREQ, IOERR, LENGERR, NOSPOOL

Description

This command obtains the next record from the system spooler. You must use the NOHANDLE option, otherwise you get an APST abend.

Options

INTO(data-area)

specifies the data area into which variable-length data will be transferred. It is a receiver field.

MAXLENGTH(data-value)

specifies the maximum length of data transferred as a fullword binary number. This is set by the user on input. It is optional, and if it is omitted CICS will use the length of the data area. Length cannot be more than 32767 bytes. This is the maximum size of a VSE/POWER record.

TOLENGTH(data-area)

specifies the length of data transferred as a fullword binary number. This is set by CICS on input. It is optional, and if it is omitted the user will not be notified of the actual length of the data received.

TOKEN(data-area)

This is the CICS-allocated token used to identify a report. It is an 8-byte value aligned on a fullword boundary. It is a receiver on SPOOLOPEN and a sender on all other commands.

Conditions

Note: RESP2 values of 256 and over are internal, except with the IOERR response. If a RESP2 value greater than 256 is returned with any RESP value but IOERR, contact your IBM support center.

ENDFILE

Indicates that all data for the current spool file being SPOOLREAD has been retrieved.

Default action: terminate the task abnormally.

INVREQ

RESP2 values:

- 1 Unsupported language (RPG II).
- 4 Attempted SPOOLREAD without previous valid SPOOLOPEN, or attempted SPOOLREAD on an output spool file (invalid token).
- 5 Attempted SPOOLREAD with a token that does not match a current token.

Default action: terminate the task abnormally.

IOERR

Indicates an input/output error. The RESP2 codes show the decimal representation of the hexadecimal POWER return and feedback codes documented in the PWRSPPL copybook. For example, a RESP2 value of 2057 is equivalent to a POWER return code of X'0809', which means "INVALID USER/REMOTE ID".

LENGERR

RESP2 values:

- 1 Insufficient buffer space to SPOOLREAD a record.

Default action: terminate the task abnormally.

NOSPOOL

RESP2 values:

- 1 The VSE/POWER interface is not available. No subsystem is present.
- 2 The VSE/POWER interface is not available. No report controller interface is present.

Default action: terminate the task abnormally.

SPOOLWRITE

Write a spool report.

```

  >>> SPOOLWRITE [FLENGTH(data-value)] FROM(data-area) [LINE] [PAGE] TOKEN(data-area) <<<

```

Conditions: INVREQ, IOERR, LENGERR, NOSPACE, NOSPOOL

Description

This command writes data to a spool report. You must use the NOHANDLE option, otherwise you get an APST abend.

Options

FLENGTH(data-value)

specifies the length of data transferred as a fullword binary number. This is set by the user on output. It is optional, and if it is omitted CICS will use the length of the data area.

FROM(data-area)

specifies the data area from which variable length data will be transferred. The data itself is not altered in any way by CICS. FROM is a sender field.

LINEPAGE

specifies the format of the data to be sent.

Note: Structured fields and line data cannot be combined within the data to be written to the report, because this will cause unpredictable results when the report is printed.

TOKEN(data-area)

This is the CICS-allocated token used to identify a report. It is an 8-byte value aligned on a fullword boundary. It is a receiver on SPOOLOPEN and a sender on all other commands.

Conditions

Note: RESP2 values of 256 and over are internal, except with the IOERR response. If a RESP2 value greater than 256 is returned with any RESP value but IOERR, contact your IBM support center.

INVREQ

RESP2 values:

- 1 Unsupported language (RPG II).
- 2 Attempted SPOOLWRITE with a spool token that does not match the token of a currently open spool file.
- 3 Attempted SPOOLWRITE without previously doing a valid SPOOLOPEN.
- 4 Attempted SPOOLWRITE to an input spool file (invalid token).

Default action: terminate the task abnormally.

IOERR

Indicates an input/output error. The RESP2 codes show the decimal representation of the hexadecimal POWER return and feedback codes documented in the PWRSPPL copybook. For example, a RESP2 value of 2057 is equivalent to a POWER return code of X'0809', which means "INVALID USER/REMOTE ID."

LENGERR

RESP2 values:

- 2 Length of SPOOLWRITE for a NOCC type spool file exceeds the maximum allowed (32 759 bytes).
- 3 Length of SPOOLWRITE for an ASA or MCC type spool file exceeds the maximum allowed (32 760 bytes).
- 4 Invalid SPOOLWRITE length. FLENGTH must be greater than 0.

Default action: terminate the task abnormally.

NOSPACE

Indicates that there is no space on the POWER spool file to add records. If you do not have the report

controller installed, spool files that are open for input are deleted by VSE/POWER. If you have the report controller installed, reports that are open for input with the LOGICAL recovery option are deleted by VSE/POWER, and reports that are open for input with the PHYSICAL recovery option are kept.

Default action: terminate the task abnormally.

NOSPOOL

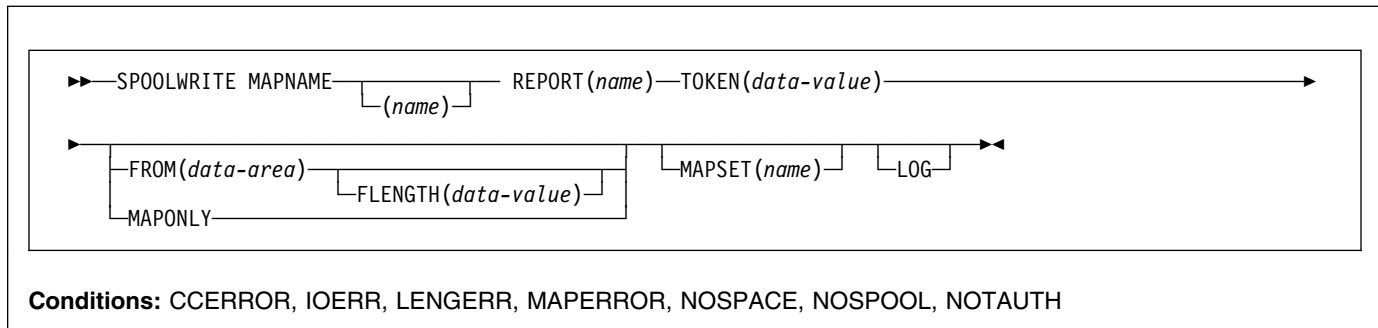
RESP2 values:

- 1** The VSE/POWER interface is not available. No subsystem is present.
- 2** The VSE/POWER interface is not available. No report controller interface is present.

Default action: terminate the task abnormally.

SPOOLWRITE MAPNAME

Write to a MAP report.



Description

SPOOLWRITE MAPNAME is a function of the report controller facility. For more information about the report controller, see the *CICS Report Controller User's Guide* and the *CICS Report Controller Planning Guide*.

Options

FLENGTH(*data-value*)

mandatory only for a COBOL program writing a non-MAP report. Specifies the length (signed binary fullword) of the data supplied with a SPOOLWRITE REPORT command.

- If the report is in ASA, MCC, NOCC, or JCL format, the number of lines to write to the report is calculated by dividing the LINELENGTH or its default into the length of the data specified by the FLENGTH option. Any remainder from the division is considered a line and is padded with blanks to full length before being written to the report.

For ASA and MCC reports, the application programmer must ensure correct carriage control characters for all lines supplied in the command.

- If the report is in ESC, MAP, SCS (with NOCONV), or T3270 (with NOCONV) format, the full length of the data is considered one unit and is written as one record. The maximum record size is 32767.

If the FLENGTH option is omitted, the following apply:

- If the report is in ASA, MCC, NOCC, JCL, T3270, SCS, or ESC format, the FLENGTH option is implicitly calculated as the length of the data field specified in the FROM option. This length is resolved at program compilation time and cannot be dynamically changed during program execution.
- If the report is in MAP format, the length of the map is calculated by scanning the physical map. If the

mapset cannot be loaded from the library or the map cannot be located within the mapset, the MAPERROR condition occurs.

FROM(*data-area*)

specifies the field containing the data to be written. For ASA, MCC, NOCC, and JCL format reports, the data area can contain any number of lines. The report controller calculates the number of lines by dividing LINELENGTH or its default into FLENGTH option. Any lines that are not filled are padded with blanks before being written.

If FLENGTH is omitted, CICS calculates the length of the data area. It is the responsibility of the application programmer to ensure correct carriage control characters in all lines supplied. If an incorrect carriage control character is encountered, the CCERROR condition occurs.

The characters used are 1–32767 for ESC, MAP, SCS (with NOCONV), and T3270 (with NOCONV). For ASA, MCC, NOCC, and JCL there is no upper limit.

LOG

specifies that a log report is to be written to. A log report is defined as a report that can be written to by a number of applications concurrently. A log report stays open until it is closed explicitly, or it is closed implicitly at CICS termination.

A log report can be in any format.

MAPNAME(*name*)

specifies the name (1–7 characters) of the BMS map to be used.

If MAPNAME is specified on a SPOOLWRITE command, the value specified at SPOOLOPEN is used.

If MAPNAME is specified on a SPOOLWRITE command, and no value is specified at SPOOLOPEN, the MAPERROR condition occurs.

The valid characters that can be used in the destination map name are as follows:

A through Z
0 through 9
@ \$ #

The first character must not be numeric.

Note: Suffixes are not supported by the report controller.

MAPONLY

specifies that only the map text is to be output. If you specify this option, you cannot specify FROM.

MAPSET(*name*)

specifies the name (1-7 characters) of the BMS mapset that contains the MAPNAME to be used for a MAP format report.

If the MAPSET option is not specified, the mapset name is assumed to be the same as the MAPNAME.

The valid characters that can be used in the destination mapset name are as follows:

A through Z
0 through 9
@ \$ #

The first character must not be numeric.

Note: Suffixes are not supported by the report controller.

REPORT(*name*)

specifies the name (2–8 characters) of the report. This option is required for every command. The name must match the name specified in the SPOOLOPEN that used the same token. Otherwise, the NAMEERROR condition occurs.

The valid characters that can be used in the report name are as follows:

A through Z
0 through 9
- . / @ \$ #

The first character must not be numeric.

Do **not** use any of the following names, because you will not be able to manipulate them using CEMS or CEOS, or using POWER operator commands:

ALL
DEV
FREE
HOLD
LOCAL
PNET
RJE
STATUS.

TOKEN(*data-value*)

used by CICS to identify the report during creation. The option is required on all commands. You must specify an 8-character area that contains the value returned after SPOOLOPEN (except for LOG reports, when a token area must be supplied but the returned value is not required).

Conditions

CCERROR

occurs when an incorrect control character has been specified.

Default action: terminate the task abnormally.

IOERR

occurs when an error response is returned from POWER. RESP2 contains the POWER return and feedback codes in the format **rrffvvvv** where:

rr IJBXRETC.
ff IJBXREAS.
vv offset 4 from IJBXRUSR (PXPRETCD).
vv offset 5 from IJBXRUSR (PXPFBKCD).

IJBXRETC and IJBXREAS are detailed in the DSECT generated by MAPXPCCB.

PXPRETCD and PXPFBKCD are detailed in the DSECT PXPUSER generated as part of PWR SPL TYPE=MAP.

Default action: terminate the task abnormally.

LENGERR

RESP2 values:

- 2 Length of SPOOLWRITE for ESC, BMS, SCS(NOCONV), or 3270(NOCONV) type report is greater than the maximum allowable 32767 bytes
- 4 Incorrect length for SPOOLWRITE. Length must be greater than 0.

Default action: terminate the task abnormally.

MAPERROR

occurs when there is an error during execution of a SPOOLWRITE MAPNAME command.

RESP2 values:

- 1 No mapset name specified for a report opened without a mapset name
- 2 No map name specified for a report opened without a map name
- 3 Mapset cannot be found.
- 4 Map name specified not found in specified mapset
- 5 No data fields in the specified map
- 6 GETMAIN for the specified map set failed.

Default action: terminate the task abnormally.

SPOOLWRITE MAPNAME

NOSPACE

occurs when the spool files are 90% full. If you continue creating reports without providing more space, IOERR occurs, and you will be unable to close your report.

Default action: terminate the task abnormally.

NOSPOOL

occurs when part, or all, of the POWER interface is unavailable.

RESP2 values:

- 1** No subsystem present
- 2** No report controller interface present
- 6** Interface being terminated
- 7** Interface not active
- 8** Interface being disabled
- 12** Interface has been stopped

Default action: terminate the task abnormally.

NOTAUTH

occurs when a resource security check has failed. The reasons for the failure are the same as for abend code AEY7, as listed in the *VSE/ESA Messages and Codes Volume 3* manual.

Default action: terminate the task abnormally.

SPOOLWRITE REPORT

Write to a report (not MAP).

```

▶▶ SPOOLWRITE REPORT (name) —TOKEN (data-value) —FROM (data-area) —
                                     FLENGTH (data-value) —
                                     LOG ▶▶

```

Conditions: CCERROR, INVREQ, IOERR, LENGERR, NAMEERROR, NOSPSPACE, NOSPOOL, NOTAUTH

Description

SPOOLWRITE REPORT is a function of the report controller facility. For more information about the report controller, see the *CICS Report Controller User's Guide* and the *CICS Report Controller Planning Guide*.

Options

FLENGTH(data-value)

mandatory only for a COBOL program writing a non-MAP report. Specifies the length (signed binary fullword) of the data supplied with a SPOOLWRITE REPORT command.

- If the report is in ASA, MCC, NOCC, or JCL format, the number of lines to write to the report is calculated by dividing the LINELENGTH or its default into the length of the data specified by the FLENGTH option. Any remainder from the division is considered a line and is padded with blanks to full length before being written to the report.
For ASA and MCC reports, the application programmer must ensure correct carriage control characters for all lines supplied in the command.
- If the report is in ESC, MAP, SCS (with NOCONV), or T3270 (with NOCONV) format, the full length of the data is considered one unit and is written as one record. The maximum record size is 32767.

If the FLENGTH option is omitted, the following applies:

- If the report is in ASA, MCC, NOCC, JCL, T3270, SCS, or ESC format, the FLENGTH option is implicitly calculated as the length of the data field specified in the FROM option. This length is resolved at program compilation time and cannot be dynamically changed during program execution.
- If the report is in MAP format, the length of the map is calculated by scanning the physical map. If the mapset cannot be loaded from the library or the map cannot be located within the mapset, the MAPERROR condition occurs.

FROM(data-area)

specifies the field containing the data to be written. For ASA, MCC, NOCC, and JCL format reports, the data area can contain any number of lines. The report controller calculates the number of lines by dividing LINELENGTH or its default into FLENGTH option. Any lines that are not filled are padded with blanks before being written.

If FLENGTH is omitted, CICS calculates the length of the data area. It is the responsibility of the application programmer to ensure correct carriage control characters in all lines supplied. If an incorrect carriage control character is encountered, the CCERROR condition occurs.

The characters used are 1–32767 characters for ESC, MAP, SCS (with NOCONV), and T3270 (with NOCONV). For ASA, MCC, NOCC, and JCL there is no upper limit.

LOG

specifies that a log report is to be written to. A log report is defined as a report that can be written to by a number of applications concurrently. A log report stays open until it is closed explicitly, or it is closed implicitly at CICS termination.

A log report can be in any format.

REPORT(name)

specifies the name (2–8 characters) of the report. This option is required for every command. The name must match the name specified in the SPOOLOPEN that used the same token. Otherwise, the NAMEERROR condition occurs.

The valid characters that can be used in the report name are as follows:

```

A through Z
0 through 9
- . / @ $ #

```

The first character must not be numeric.

Do **not** use any of the following names, because you will not be able to manipulate them using CEMS or CEOS, or using POWER operator commands:

SPOOLWRITE REPORT

ALL
DEV
FREE
HOLD
LOCAL
PNET
RJE
STATUS

TOKEN(*data-value*)

used by CICS to identify the report during creation. The option is required on all commands. You must specify an 8-character area that contains the value returned after SPOOLOPEN (except for LOG reports, when a token area must be supplied but the returned value is not required).

Conditions

CCERROR

occurs when an incorrect control character has been specified.

Default action: terminate the task abnormally.

INVREQ

occurs when an invalid token is specified.

Default action: terminate the task abnormally.

IOERR

occurs when an error response is returned from POWER. RESP2 contains the POWER return and feedback codes in the format **rrffvvvv** where:

rr IJBXRETC.

ff IJBXREAS.

vv offset 4 from IJBXRUSR (PXPRETCD).

vv offset 5 from IJBXRUSR (XPFBKCD).

IJBXRETC and IJBXREAS are detailed in the DSECT generated by MAPXPCCB.

PXPRETCD and XPFBKCD are detailed in the DSECT PXPUSER generated as part of PWRSPPL TYPE=MAP.

Default action: terminate the task abnormally.

LENGERR

RESP2 values:

- 2** Length of SPOOLWRITE for ESC, BMS, SCS(NOCONV), or 3270(NOCONV) type report is greater than the maximum allowable 32 767 bytes
- 4** Incorrect length for SPOOLWRITE. Length must be greater than 0.

Default action: terminate the task abnormally.

NAMEERROR

occurs when an application specifies the wrong report name when issuing the SPOOLWRITE REPORT command.

Default action: terminate the task abnormally.

NOSPACE

occurs when the spool files are 90% full. If you continue creating reports without providing more space, IOERR occurs, and you will be unable to close your report.

Default action: terminate the task abnormally.

NOSPOOL

occurs when part, or all, of the POWER interface is unavailable.

RESP2 values:

- 1** No subsystem present
- 2** No report controller interface present
- 6** Interface being terminated
- 7** Interface not active
- 8** Interface being disabled
- 12** Interface has been stopped

Default action: terminate the task abnormally.

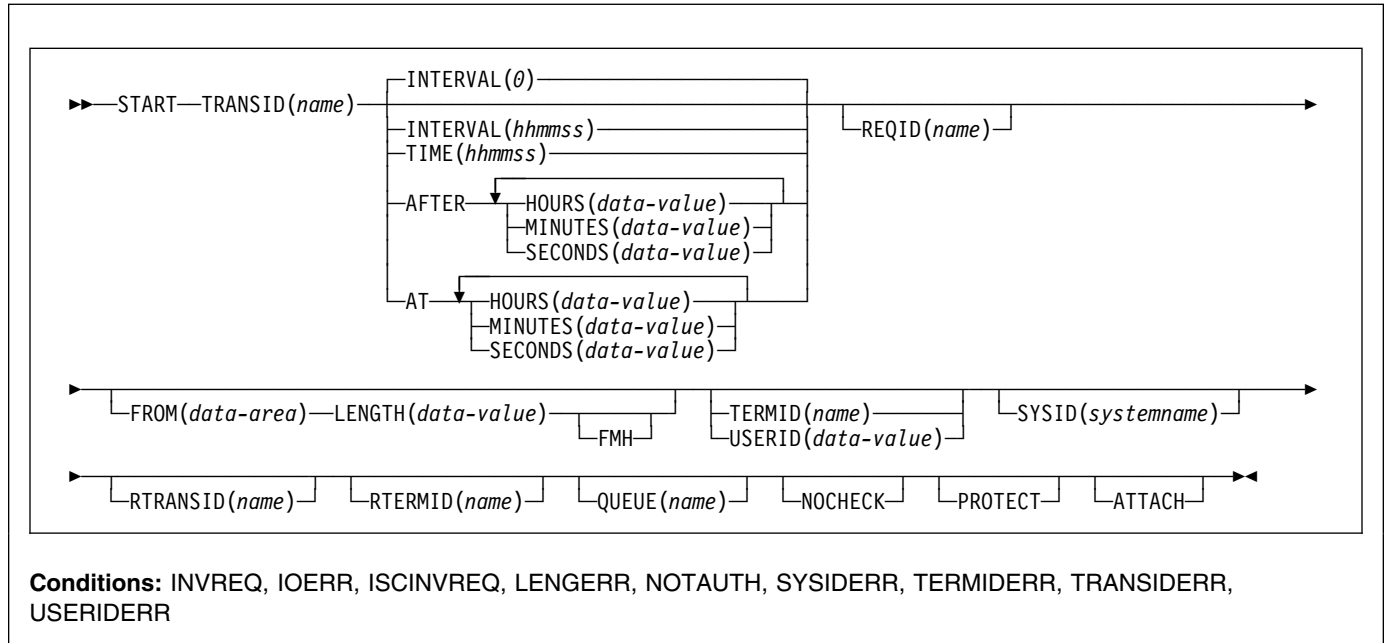
NOTAUTH

occurs when a resource security check has failed. The reasons for the failure are the same as for abend code AEY7, as listed in the *VSE/ESA Messages and Codes Volume 3* manual.

Default action: terminate the task abnormally.

START

Start task at a specified time.



Note for dynamic transaction routing

Using START if later CANCELED by another task, or if the started transaction uses RETRIEVE WAIT, could create inter-transaction affinities that adversely affect the use of dynamic transaction routing. See the *CICS Application Programming Guide*

for more information about transaction affinities.

Description

START starts a task, on a local or remote system, at a specified time. The time is specified by INTERVAL, AFTER, AT or TIME. See the section about expiration times in the *CICS Application Programming Guide*.

The starting task may pass data to the started task. The starting task may also specify a terminal to be used by the started task as its principal facility.

The default is INTERVAL(0), but for C the default is AFTER HOURS(0) MINUTES(0) SECONDS(0).

Note that **CEDF** is an exception to the START command and is not valid as a TRANSID name. You should therefore not attempt to start CEDF in this way.

You can use the RTRANSID, RTERMID, and QUEUE options to pass further data to the started task. These options can contain arbitrary data values whose meanings depend on what you have specified in the started and starting tasks. One possible way of using them is in the

following situation. One task can start a second task, passing it a transaction name and a terminal name to be used when the second task starts a third task. The first task may also pass the name of a queue to be accessed by the second task.

One or more constraints have to be satisfied before the transaction to be executed can be started, as follows:

- The specified interval must have elapsed or the specified expiration time must have been reached. (For more information, see the *CICS Application Programming Guide*.) The INTERVAL or AFTER options should be specified when a transaction is to be executed on a remote system; this avoids complications arising when the local and remote systems are in different time zones.
- If the TERMID option is specified, the named terminal must exist and be available. If the named terminal does not exist when the time interval expires, the START is discarded.
- If the PROTECT option is specified, the starting task must have taken a successful syncpoint. This option,

START

coupled to extensions to system tables, reduces the exposure to lost or duplicated data caused by failure of a starting task.

- If the transaction to be executed is on a remote system, the format of the data must be declared to be the same as that at the local system. This is done using the RDO CONNECTION options DATASTREAM and RECORDFORMAT. For CICS-CICS, these are always the default values.

Execution of a START command naming a transaction in the local system cancels any outstanding POST commands executed by the starting task.

START commands can be queued by specifying the LOCALQ(YES) option on the RDO TRANSACTION resource definition, as described in the *CICS Resource Definition Guide*.

Passing data by interval control

If data is to be passed by interval control (using the FROM option), it is queued on a temporary storage queue. The REQID option allows you to specify the name of the temporary storage queue to be used. This identifier may be recoverable (in temporary storage terms) or nonrecoverable. The *CICS Resource Definition Guide* describes how to define recoverable temporary storage queues.

If you also specify the PROTECT option, the temporary storage queue identified by the REQID option should be defined as recoverable. If you do not specify the PROTECT option, the temporary storage queue should not be defined as recoverable. Unpredictable results can occur if these rules are not followed (see the *CICS Recovery and Restart Guide*).

If you specify the FROM and not the REQID option, a default 'DF' prefix temporary storage queue is used. The same rules apply as above; only specify the PROTECT option if you define the 'DF' prefix temporary storage queues as recoverable.

Error checking and performance considerations

The NOCHECK option specifies that no response (to execution of the START command) is expected by the starting transaction. For START commands naming tasks to be started on a local system, error conditions are returned; error conditions are not returned for tasks to be started on a remote system. The NOCHECK option allows CICS to improve performance when the START command has to be shipped to a remote system; it is also a prerequisite if the shipping of the START command is queued pending the establishing of links to the remote system.

Starting tasks without terminals

If the task to be started is not associated with a terminal, each START command results in a separate task being started. This happens regardless of whether or not data is passed to the started task. The following examples show how to start a specified task, not associated with a terminal, in one hour:

```
EXEC CICS START
      TRANSID('TRNL')
      INTERVAL(10000)
      REQID('NONGL')
:
EXEC CICS START
      TRANSID('TRNL')
      AFTER HOURS(1)
      REQID('NONGL')
:
```

Starting tasks with terminals but without data

Only one task is started if several START commands, each specifying the same transaction and terminal, expire at the same time or before the terminal is available.

The following examples show how to request initiation of a task associated with a terminal. Because no request identifier is specified in these examples, CICS assigns one and returns it to the application program in the EIBREQID field of the EXEC interface block.

```
EXEC CICS START
      TRANSID('TRN1')
      TIME(185000)
      TERMID('STA5')
:
EXEC CICS START
      TRANSID('TRN1')
      AT HOURS(18) MINUTES(50)
      TERMID('STA5')
:
```

Starting tasks with terminals and data

Data is passed to a started task if one or more of the FROM, RTRANSID, RTERMID, and QUEUE options is specified. Such data is accessed by the started task by using a RETRIEVE command.

It is possible to pass many data records to a new task by issuing several START commands, each specifying the same transaction and terminal.

Execution of the first START command ultimately causes the new task to be started and allows it to retrieve the data

specified on the command. The new task is also able to retrieve data specified on subsequently executed START commands that expire before the new task is terminated. If the transaction has been defined as restartable (by defining the transaction using the RDO TRANSACTION option RESTART(YES)) and such data has not been retrieved before the new task is terminated, another new task is started and is able to retrieve the outstanding data. If the subsequent new task fails to retrieve the outstanding data, a third task will be started and so on, up to a maximum of five times, after which the data is discarded. If the transaction has not been defined as restartable, no new task is initiated and the data is discarded.

The following examples show how to start a task associated with a terminal and pass data to it:

```
EXEC CICS START
  TRANSID('TRN2')
  TIME(173000)
  TERMID('STA3')
  REQID(DATAREC)
  FROM(DATAFLD)
  LENGTH(100)
:
EXEC CICS START
  TRANSID('TRN2')
  AT HOURS(17) MINUTES(30)
  TERMID('STA3')
  REQID(DATAREC)
  FROM(DATAFLD)
  LENGTH(100)
:
```

When using the C language, you are recommended to use the AFTER/AT HOURS, MINUTES, and SECONDS options as C does not provide a packed decimal data type. You may use INTERVAL or TIME, but if the value specified is **not** an integer constant, the application is responsible for ensuring that the value passed to CICS is in packed decimal format.

START failures without exception conditions

There are some circumstances in which a START command is executed without error, but the started task never takes place:

- When the transaction or its initial program is disabled at the time CICS attempts to create the task.
- When the START specifies a terminal and an expiration time, and the terminal is not defined (and cannot be located by the XICTENF or XALTENF global user exits) at expiration time. See the *CICS Customization Guide* for more information about the CICS global user exits.
- When the START specifies a terminal that is not defined (and cannot be located by the XICTENF or XALTENF exits) at the time CICS attempts to create the task.

These exposures result from the delay between the execution of the START and the time of task creation. Even when the START is immediate, CICS may delay creating the task, either because the required terminal is not free or because of other system constraints.

You can use INQUIRE commands to ensure that the transaction and program are enabled at the time of the START command, but either may become disabled before task creation.

You get a TERMIDERR condition if the requested terminal does not exist at the time of the START, but if it is deleted subsequently, as occurs if the user logs off, your START request is discarded with the terminal definition.

Options

AFTER

specifies the interval of time that is to elapse before the new task is started.

There are two ways to enter the time under AFTER and AT.

1. A combination of at least two of HOURS(0–99), MINUTES(0–59), and SECONDS(0–59). HOURS(1) SECONDS(3) would mean one hour and three seconds (the minutes default to zero).
2. As one of HOURS(0–99), MINUTES(0–5999), or SECONDS(0–359999). HOURS(1) means one hour. MINUTES(62) means one hour and two minutes. SECONDS(3723) means one hour, two minutes, and three seconds.

AT specifies the time at which the new task is to be started. For the ways to enter the time, see the AFTER option.

FMH

specifies that the user data to be passed to the started task contains function management headers. This is not valid for LUTYPE2 or LUTYPE3 terminals.

FROM(*data-area*)

specifies the data to be stored for a task that is to be started at some future time.

HOURS(*data-value*)

specifies a fullword binary value in the range 0–99. This is a suboption of the AFTER and AT options. For its use and meaning, see the AFTER option.

INTERVAL(*hhmmss*)

specifies the expiration time as an interval of time that is to elapse from the time at which the START command is issued. The **mm** and **ss** are each in the range 0–59. The time specified is added to the current clock time by CICS when the command is executed, to calculate the expiration time.

LENGTH(*data-value*)

specifies a halfword binary data value that is the length of the data to be stored for the new task.

START

MINUTES(*data-value*)

specifies as a fullword binary value the number of minutes for use in conjunction with AFTER or AT. The value must be in the range 0 through 59 if HOURS or SECONDS is also specified, or in the range 0 through 5999 otherwise. This is a suboption of the AFTER and AT options. For its use and meaning, see the AFTER option.

NOCHECK

specifies that, for a remote system, CICS should improve performance of the START command by providing less error checking and slightly less function. For more information, see the section about improving the performance of intersystem START requests in the *CICS Intercommunication Guide*.

PROTECT

specifies that the new task is not started until the starting task has taken a syncpoint. If the starting task abends before the syncpoint is taken, the request to start the new task is canceled. If the REQID option is also specified, the request identifier should be a name defined as recoverable to temporary storage. If the started transaction is remote, PROTECT specifies that it must not be scheduled until the local transaction has successfully completed a syncpoint. For more information about the PROTECT option with remote transactions, see the *CICS Intercommunication Guide*.

QUEUE(*name*)

specifies the name (1–8 characters) of a temporary storage queue that may be used by the started transaction.

If you are also specifying REQID, make sure that the name of the REQID and the name of the QUEUE are not the same.

REQID(*name*)

specifies a name (1–8 characters), which must be unique, to identify a command. This option can be used when another task is to be provided with the capability of canceling an unexpired command.

If this option is omitted, CICS generates a unique request identifier in the EIBREQID field of the EXEC interface block, unless the NOCHECK option is specified, in which case field EIBREQID is set to nulls and cannot be used subsequently to cancel the START command.

If you include any of the data options (FROM, RTERMID, RTRANSID or QUEUE), the data is stored in a TS queue using the REQID name specified (or CICS generated) as the identifier. The queue record thus identified must be local to the CICS system where the START command is processed. The START command is processed on the system identified by the SYSID option or, if the SYSID option is omitted, on the system associated with the TRANSID option.

You cannot specify a REQID value that already exists for another start request that has not yet expired.

RTERMID(*name*)

specifies a value (1–4 characters), for example a terminal name, that may be retrieved when the transaction, specified in the TRANSID option in the START command, is started.

When retrieved, the value may be used in the TERMID option of a subsequent START command.

RTRANSID(*name*)

specifies a value (1–4 characters), for example a transaction name, that may be retrieved when the transaction, specified in the TRANSID option in the START command, is started.

When retrieved, the value may be used in the TRANSID option of a subsequent START command.

SECONDS(*data-value*)

specifies a fullword binary value in the range 0–59, when HOURS or MINUTES are also specified, or 0–359 999 when SECONDS is the only option specified. This is a suboption of the AFTER and AT options. For its use and meaning, see the AFTER option.

SYSID(*systemname*)

specifies the name of the system to which the request is directed.

TERMID(*name*)

specifies the symbolic identifier (1–4 alphanumeric characters) of the principal facility associated with a transaction to be started as a result of a START command. This principal facility can be either a terminal (the usual case) or an APPC session. Where an APPC session is specified, the connection (or modeset) name is used instead of a terminal identifier. This option is required when the transaction to be started must communicate with a terminal; it should be omitted otherwise.

The terminal identifier must be defined as either a local or a remote terminal on the system in which the START command is executed, **when the start of the transaction takes effect**.

TIME(*hhmmss*)

specifies the time when a new task should be started.

When using the C language, you are recommended to use the AFTER/AT HOURS, MINUTES, and SECONDS options as C does not provide a packed decimal data type. You may use TIME, but if the value specified is **not** an integer constant, the application is responsible for ensuring that the value passed to CICS is in packed decimal format.

TRANSID(*name*)

specifies the symbolic identifier (1–4 characters) of the transaction to be executed by a task started as the result of a START command.

If SYSID is specified, and names a remote system, the transaction is assumed to be on that system irrespective

of whether or not the transaction definition is defined as remote in the PCT. Otherwise the transaction definition is used to find out whether the transaction is on a local or a remote system.

USERID(*data-value*)

Specifies the userid under whose authority the started transaction is to run, if the started transaction is not associated with a terminal (that is, when **TERMID** is not specified). This is referred to as *userid1*.

If you omit both **TERMID** and **USERID**, CICS uses instead the userid under which the transaction that issues the **START** command is running. This is referred to as *userid2*.

By using either *userid1* or *userid2* CICS ensures that a started transaction always runs under a valid userid, which must be authorized to all the resources referenced by the started transaction.

CICS performs a surrogate security check against *userid2* to verify that this user is authorized to *userid1*. If *userid2* is not authorized, CICS returns a **NOTAUTH** condition. The surrogate check is not done here if **USERID** is omitted.

Conditions

INVREQ

RESP2 values:

- 4 The value specified in the **HOURS** option is out of range.
- 5 The value specified in the **MINUTES** option is out of range.
- 6 The value specified in the **SECONDS** option is out of range.
- 18 A **USERID** is specified and the CICS external security manager interface is not initialized.

Also occurs (**RESP2** not set) in any of the following situations:

- The **START** command is not valid for processing by CICS.
- Values specified in the **INTERVAL** option are out of range.

Default action: terminate the task abnormally.

IOERR

occurs in any of the following situations:

- An input/output error occurred during a **START** operation.
- A **START** operation attempts to write to a temporary storage queue when the **DFHTEMP** dataset is full.
- A **START** operation uses a **REQID** name that already exists. This condition only occurs when the **FROM** option is also used.

Default action: terminate the task abnormally.

ISCVNREQ

occurs when the remote system indicates a failure that does not correspond to a known condition.

Default action: terminate the task abnormally.

LENGERR

occurs if **LENGTH** is not greater than zero.

Default action: terminate the task abnormally.

NOTAUTH

RESP2 values:

- 7 A resource security check fails on **TRANSID** (name).
- 9 A surrogate user security check fails on **USERID** (name).

The security access capabilities of the transaction that issued the command do not allow the command to be performed with the value specified in the **USERID** option. The security access capabilities of the transaction have been established by the external security manager according to user security, and whether link security or the execution diagnostic facility (EDF) have been in use.

Default action: terminate the task abnormally.

SYSIDERR

occurs when the **SYSID** option specifies a name that is neither the local system nor a remote system (made known to CICS by defining a **CONNECTION**). **SYSIDERR** also occurs when the link to the remote system is known but unavailable. This condition may not be raised if the global user exit **XISLCLQ** is enabled (see the *CICS Customization Guide* for programming information).

Default action: terminate the task abnormally.

TERMIDERR

occurs if the terminal identifier in a **START** command cannot be found in the terminal control table.

Default action: terminate the task abnormally.

TRANSIDERR

occurs if the transaction identifier specified in a **START** command cannot be found in the program control table.

Default action: terminate the task abnormally.

USERIDERR

RESP2 values:

- 8 The specified **USERID** is not known to the external security manager.
- 10 The external security manager is in a state such that CICS cannot determine whether a specified **USERID** is valid.

Default action: terminate the task abnormally.

START ATTACH

START ATTACH

Start a task immediately.

```
▶▶—START ATTACH—TRANSID(name)—FROM(data-area)—LENGTH(data-value)—◀◀
```

Conditions: INVREQ, LENGERR, NOTAUTH, TRANSIDERR

Description

START ATTACH starts a non-terminal task immediately in the local CICS region.

The attached task has a STARTCODE of U and cannot be cancelled, so EIBREQID is set to nulls.

ATTACH allows a START issued in a PLTPI program to take effect before initialization has completed.

The starting task may pass data to the started task using the FROM option.

Starting tasks with data

If data is to be passed it is not written to a temporary storage queue; only its address is passed.

The attached task retrieves data in the normal way. The task issuing the START must ensure that the data is valid when it is retrieved, either by synchronizing its execution with the attached task, or by placing the data in shared storage.

Each START ATTACH command results in a separate task being started, whether or not data is passed to the started task. The following example shows how to start a specified task, and pass data to it:

```
EXEC CICS START ATTACH
          TRANSID('TRNL')
          FROM(DATAFLD)
          LENGTH(100)
          :
```

Options

FROM(*data-area*)

specifies the data to be passed to a started task.

LENGTH(*data-value*)

specifies a halfword binary data value that is the length of the data to be passed to a started task.

TRANSID(*name*)

specifies the symbolic identifier (1–4 characters) of the transaction to be executed by a task started as the result of a START ATTACH command.

Conditions

INVREQ

RESP2 values:

- 11 An attempt was made to route a START ATTACH request.
- 12 A START ATTACH request has failed.

Default action: terminate the task abnormally.

LENGERR

occurs if LENGTH is not greater than zero.

Default action: terminate the task abnormally.

NOTAUTH

RESP2 values:

- 7 A resource security check fails on TRANSID (*name*).

Default action: terminate the task abnormally.

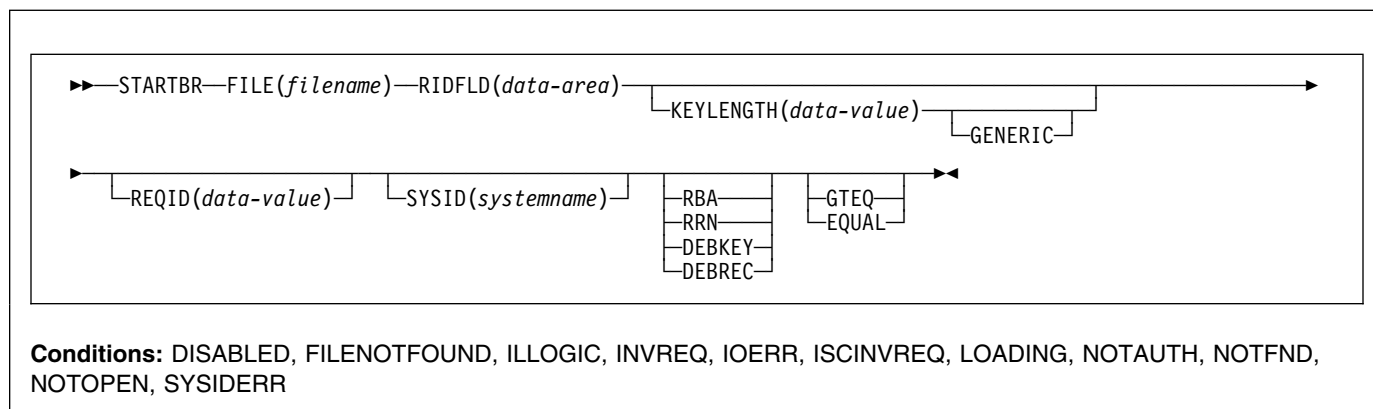
TRANSIDERR

occurs if the transaction identifier specified in a START command has not been defined to CICS.

Default action: terminate the task abnormally.

STARTBR

Start browse of a file.



Description

STARTBR specifies the record in a file, or in a data table, on a local or a remote system, where you want the browse to start. No records are read until a READNEXT command (or, for VSAM and tables, a READPREV command) is executed.

A browse operation, where direct means browse of the base data set (using the primary key), may be:

- A direct browse of a key sequenced data set (KSDS or data-table) by record key.
- A direct browse of an entry sequenced data set (ESDS) by relative byte address (RBA).
- A direct browse of a relative record data set (RRDS) by relative record number (RRN).
- A browse of a key sequenced data set (KSDS) using an alternate index path.
- A browse of an entry sequenced data set (ESDS) using an alternate index path. In this case, an ESDS is browsed by key in the same way as a KSDS. Some of the options that are not valid for a direct ESDS browse are valid for an alternate index browse.
- A browse of a KSDS by RBA.

The options specified on the STARTBR command define the characteristics that apply throughout the subsequent browse operation. Specifically, if GENERIC or GTEQ are specified, they are used not only when determining the starting point of the browse, but also whenever the value of RIDFLD is changed before issuing a READNEXT command.

If you specify the RBA option, it applies to every READNEXT or READPREV command in the browse, and causes CICS to return the relative byte address of each retrieved record.

None of these options can be changed during a browse, except by means of the RESETBR command.

If a STARTBR request specifies the precise key at which the browse is to start (that is, it specifies a full key and the EQUAL keyword) the record returned on the following READNEXT (or READPREV) may not be the same as the record specified by the STARTBR for a file opened in VSAM NSR mode. This can occur because the initial record specified on the STARTBR command can be deleted by another transaction in between the STARTBR completing and a READNEXT or READPREV being issued. In VSAM LSR mode, the initial record cannot be deleted between the STARTBR and the READNEXT.

Options

DEBKEY

(blocked DAM) specifies that deblocking is to occur by key. If neither DEBREC nor DEBKEY is specified, deblocking does not occur.

DEBREC

(blocked DAM) specifies that deblocking is to occur by relative record (relative to zero). If neither DEBREC nor DEBKEY is specified, deblocking does not occur.

EQUAL

(VSAM and data table) specifies that the search is satisfied only by a record having the same key (complete or generic) as that specified in the RIDFLD option.

This option is the default field for a direct ESDS browse.

FILE(filename)

specifies the name of the file to be accessed.

If SYSID is specified, the data set to which this file refers is assumed to be on a remote system irrespective of

STARTBR

whether the name is defined in the FCT. Otherwise, the FCT entry is used to find out whether the data set is on a local or a remote system.

GENERIC

(VSAM KSDS, path or data table) specifies that the search key is a generic key whose length is specified in the KEYLENGTH option. The search for a record is satisfied when a record is found that has the same starting characters (generic key) as those specified.

GTEQ

(VSAM or data table) specifies that, if the search for a record having the same key (complete or generic) as that specified in the RIDFLD option is unsuccessful, the first record having a greater key satisfies the search.

This option is the default for directly browsing through a KSDS or an RRDS. It is not valid for directly browsing an ESDS, although it is valid for browsing through an ESDS using a path.

KEYLENGTH(*data-value*)

specifies the length (halfword binary) of the key that has been specified in the RIDFLD option, except when RBA or RRN is specified, in which case KEYLENGTH is not valid.

This option must be specified if GENERIC is specified, and it can be specified whenever a key is specified. If the length specified is different from the length defined for the data set and the operation is not generic, the INVREQ condition occurs.

The INVREQ condition also occurs if a STARTBR command specifies GENERIC, and the KEYLENGTH is not less than that specified in the VSAM definition.

If KEYLENGTH(0) is used with the object of positioning on the first record in the data set, the GTEQ option must also be specified. If EQUAL is specified either explicitly or by default with KEYLENGTH(0), the results of the STARTBR is unpredictable.

RBA

(VSAM KSDS or ESDS base data sets, or CICS-maintained data tables only, not paths) specifies that the record identification field specified in the RIDFLD option contains a relative byte address. Use this option only when browsing an ESDS base, or a KSDS base when using relative byte addresses instead of keys to identify the records.

- User-maintained data tables
- KSDS files that hold more than 4GB

REQID(*data-value*)

specifies as a halfword binary value a unique request identifier for a browse, used to control multiple browse operations on the same or different data sets. If this option is not specified, a default value of zero is assumed.

RIDFLD(*data-area*)

specifies the record identification field. The contents can be a key, a relative byte address, or relative record number (for VSAM data sets), or a block reference, physical key, and deblocking argument (for DAM data sets). For a relative byte address or a relative record number, the format of this field must be fullword binary. For a relative byte address, the RIDFLD can be greater than or equal to zero. For a relative record number, the RIDFLD can be greater than or equal to 1.

For VSAM, a full record id of X'FF's indicates that the browse is to be positioned at the end of the data set in preparation for a backwards browse using READPREV commands.

RRN

(VSAM RRDS) specifies that the record identification field specified in the RIDFLD option contains a relative record number. This option should only be used with files referencing relative record data sets.

SYSID(*systemname*)

specifies the name of the system to which the request is directed.

If you specify SYSID, and omit both RBA and RRN, you must also specify KEYLENGTH.

Conditions

Note: RESP2 values are not set for files that are on remote systems.

DISABLED

RESP2 values:

- 50** A file is disabled. A file may be disabled because:
- It was initially defined as disabled and has not since been enabled.
 - It has been disabled by a SET FILE or a CEMT SET FILE command.

Default action: terminate the task abnormally.

FILENOTFOUND

RESP2 values:

- 1** A file name referred to in the FILE option cannot be found in the FCT and SYSID has not been specified.

Default action: terminate the task abnormally.

ILLOGIC

RESP2 values: (VSAM)

- 110** A VSAM error occurs that does not fall within one of the other CICS response categories.

(See EIBRCODE in the EXEC interface block; refer to Appendix A, "EXEC interface block" on page 373 for details.)

Default action: terminate the task abnormally.

INVREQ

RESP2 values:

- 20** Browse operations are not allowed according to the file entry specification in the FCT.
- 25** The KEYLENGTH and GENERIC options are specified, and the length defined for the data set to which this file specified in the KEYLENGTH option is greater than or equal to the length of a full key.
- 26** The KEYLENGTH option is specified (but the GENERIC option is not specified), and the specified length does not equal the length defined for the data set to which this file refers.
- 33** An attempt is made to start a browse with a REQID already in use for another browse.
- 42** The KEYLENGTH and GENERIC options are specified, and the length specified in the KEYLENGTH option is less than zero.
- 44** The specified file is a user-maintained data table and the command does not conform to the format of STARTBR for a user-maintained data table (for example, RBA is specified).
- # 49** An attempt was made to browse a file that is an invalid file type.

Default action: terminate the task abnormally.

IOERR

RESP2 values:

- 120** There is an I/O error during the file control operation. An I/O error is any unusual event that is not covered by a CICS condition.

For VSAM files, IOERR normally indicates a hardware error.

(Further information is available in the EXEC interface block; refer to Appendix A, "EXEC interface block" on page 373 for details.)

Default action: terminate the task abnormally.

ISCINVREQ

RESP2 values:

- 70** The remote system indicates a failure that does not correspond to a known condition.

Default action: terminate the task abnormally.

LOADING

RESP2 values:

- 104** A STARTBR is issued to a user-maintained data table that is currently being loaded. Only reads with precise keys are allowed to a user-maintained data table during loading.

Default action: terminate the task abnormally.

NOTAUTH

RESP2 values:

- 101** A resource security check has failed on FILE(filename).

Default action: terminate the task abnormally.

NOTFND

RESP2 values:

- 80** An attempt to position on a record based on the search argument provided is unsuccessful.

NOTFND can also occur if a generic STARTBR with KEYLENGTH(0) specifies the EQUAL option.

Default action: terminate the task abnormally.

NOTOPEN

RESP2 values:

- 60** The requested file is CLOSED and UNENABLED. The CLOSED, UNENABLED state is reached after a CLOSE request has been received against an OPEN ENABLED file and the file is no longer in use. This state can also be specified as the initial state by means of the FILSTAT parameter of the DFHFCT TYPE=FILE macro, or the STATUS(ENABLED) and OPENTIME(FIRSTREF) options on the RDO FILE resource definitions.

The requested file is OPEN and in use by other transactions, but a CLOSE request against the file has been received.

This condition does not occur if the request is made either to a CLOSED, ENABLED file or to a CLOSED, DISABLED file. In the first case, the file is opened as part of executing the request. In the second case, the DISABLED condition occurs.

Default action: terminate the task abnormally.

SYSIDERR

RESP2 values:

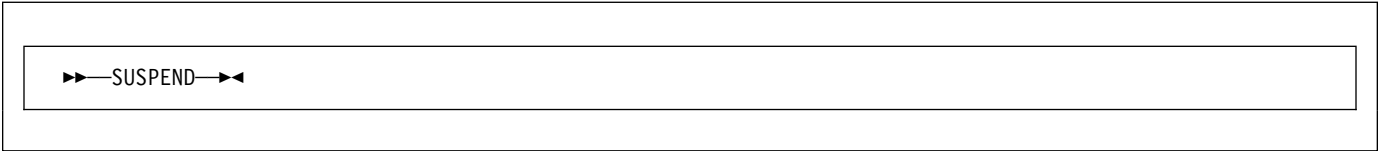
- 130** The SYSID option specifies a name that is neither the local system nor a remote system (made known to CICS by defining a CONNECTION). SYSIDERR also occurs when the link to the remote system is closed.

Default action: terminate the task abnormally.

SUSPEND

SUSPEND

Suspend a task.



Description

SUSPEND relinquishes control to a task of higher or equal dispatching priority. Control is returned to the task issuing the command as soon as no other task of a higher or equal priority is ready to be processed.

SYNCPOINT

Establish a syncpoint.

▶▶—SYNCPOINT—◀◀

Conditions: INVREQ, ROLLEDBACK

Description

SYNCPOINT divides a task (usually a long-running one) into smaller units of work. It specifies that all changes to recoverable resources made by the task since its last syncpoint are to be committed.

Note: A failure occurring during the commit phase (phase 2) of syncpoint processing does not return an error condition and the transaction is not abnormally terminated. Subsequent units of work in the transaction are allowed to continue normally. See the *CICS Recovery and Restart Guide* for further information.

Conditions

INVREQ

RESP2 values:

200 SYNCPOINT was in a program that is linked to from a remote system that has not specified the SYNCONRETURN option, or if it has been linked to locally and is defined with EXECUTIONSET=DPLSUBSET.

Default action: terminate the task abnormally.

ROLLEDBACK

occurs when a SYNCPOINT command is driven into rollback by a remote system that is unable to commit the syncpoint. All changes made to recoverable resources in the current unit of work are backed out.

Default action: terminate the task abnormally.

SYNCPOINT ROLLBACK

Back out to last syncpoint.

▶▶—SYNCPOINT—ROLLBACK—▶▶

Condition: INVREQ

Options

ROLLBACK

specifies that all changes to recoverable resources made by the task since its last syncpoint are to be backed out.

This option can be used, for example, to tidy up in a HANDLE ABEND routine, or to revoke database changes after the application program finds irrecoverable errors in its input data.

If the unit of work updates remote recoverable resources using an MRO or APPC session, the ROLLBACK option is propagated to the back-end transaction.

When a distributed transaction processing conversation is in use, the remote application program has the EIB fields EIBSYNRB, EIBERR, and EIBERRCD set. For the conversation to continue, the remote application program should execute a SYNCPOINT ROLLBACK command.

When the mirror transaction is involved in the unit of work using an MRO or APPC session, the mirror honors the rollback request, revokes changes, and then terminates normally.

This option is not supported across LUTYPE6.1 VTAM sessions to the mirror or back-end transactions. In these cases, the front-end transactions could be abended to cause the back-end transactions to back out.

Note: A failure occurring during the backout phase (phase 2) of syncpoint processing does not return an error condition and the transaction is not abnormally terminated. Subsequent units of work in the transaction are allowed to continue normally. See the *CICS Recovery and Restart Guide* for further information.

Conditions

INVREQ

RESP2 values:

- 200** SYNCPOINT ROLLBACK was in a program that is linked to from a remote system that has not specified the SYNCONRETURN option, or if it has been linked to locally and is defined with EXECUTIONSET=DPLSUBSET

Default action: terminate the task abnormally.

TERM (DL/I)

Terminate access to the PSB.



▶▶—TERM—◀◀

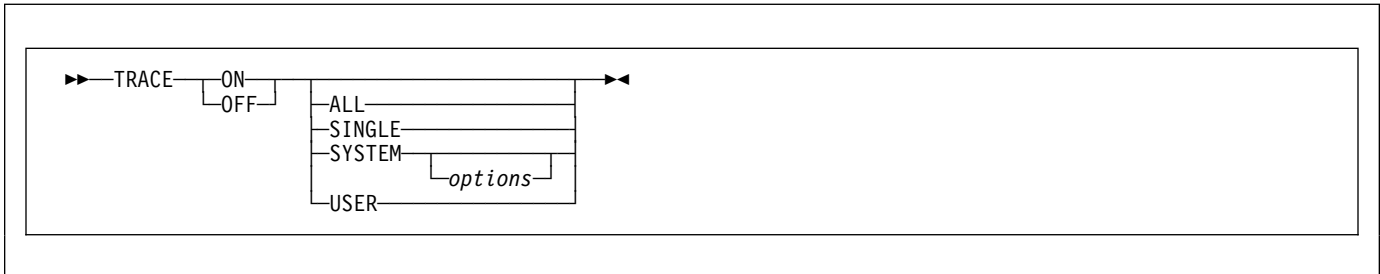
Description

The full form of the command is TERMINATE. See *DL/I DOS/VS Application Programming: High Level Programming Interfaces* for details.

TRACE

TRACE

Control the CICS trace facility.



This command is now superseded by the SET TRACEDEST and SET TRACEFLAG commands, but is still supported for compatibility with previous releases.

See the *CICS System Programming Reference* manual for further information.

UNLOCK

Release exclusive control.

```
▶▶ UNLOCK—FILE(filename)—┌TOKEN(data-area)—┐ ┌SYSID(systemname)—┐ ▶▶
```

Conditions: DISABLED, FILENOTFOUND, ILLOGIC, INVREQ, IOERR, ISCVREQ, NOTAUTH, NOTOPEN, SYSIDERR

Description

UNLOCK releases the exclusive control established in response to a read command with the UPDATE option. You use it if you retrieve a record for update, and then decide that you do not want to update the record after all. However, for a recoverable file the resource remains locked until either a syncpoint command is executed or the task is terminated. The record can be in a data set, or in a CICS or user-maintained data table, on a local or a remote system.

If an UNLOCK command does not have a token, an attempt is made to match it to either a read with the UPDATE option that also does not have a token, or to a WRITE MASSINSERT. If neither of these is found, no action is taken and a NORMAL response is returned.

Use this command to terminate a VSAM WRITE MASSINSERT operation against a VSAM file.

Options

FILE(*filename*)

specifies the name of the file to be released.

If SYSID is specified, the data set to which this file refers is assumed to be on a remote system irrespective of whether the name is defined in the FCT. Otherwise, the FCT entry is used to find out whether the data set is on a local or a remote system.

SYSID(*systemname*)

specifies the name of the system to which the request is directed.

TOKEN(*data-area*)

specifies as a fullword binary value a unique request identifier for an UNLOCK, used to associate it with a previous READ, command that specified the UPDATE option.

Conditions

Note: RESP2 values are not set for files that are on remote systems.

DISABLED

RESP2 values:

- 50** A file is disabled because it was initially defined as disabled and has not since been enabled.

A file is disabled by an EXEC CICS SET FILE or a CEMT SET FILE command.

This condition cannot occur when the UNLOCK follows a successful read for update or a VSAM WRITE MASSINSERT.

Default action: terminate the task abnormally.

FILENOTFOUND

RESP2 values:

- 1** A file name referred to in the FILE option cannot be found in the FCT and SYSID has not been specified.

Default action: terminate the task abnormally.

ILLOGIC

RESP2 values: (VSAM and CICS-maintained data tables)

- 110** A VSAM error occurs that does not fall within one of the other CICS response categories.

(See EIBRCODE in the EXEC interface block; refer to Appendix A, "EXEC interface block" on page 373 for details.)

Default action: terminate the task abnormally.

INVREQ

RESP2 values:

- 47** An unlock includes a token whose value cannot be matched against any token in use for an existing READ with the UPDATE option.
- 48** An attempt is made to function-ship a request which includes a TOKEN keyword.

UNLOCK

- # **49** An attempt was made to unlock a file that is an
invalid file type.

Default action: terminate the task abnormally.

IOERR

RESP2 values:

- 120** There is an I/O error during the file control operation. An I/O error is any unusual event that is not covered by a CICS condition.

For VSAM files, IOERR normally indicates a hardware error.

(Further information is available in the EXEC interface block; refer to Appendix A, "EXEC interface block" on page 373 for details.)

Default action: terminate the task abnormally.

ISCVREQ

RESP2 values:

- 70** The remote system indicates a failure that does not correspond to a known condition.

Default action: terminate the task abnormally.

NOTAUTH

RESP2 values:

- 101** A resource security check has failed on FILE(filename).

Default action: terminate the task abnormally.

NOTOPEN

RESP2 values:

- 60** The requested file is CLOSED and UNENABLED. The CLOSED, UNENABLED state is reached after a CLOSE request has been received against an OPEN ENABLED file and the file is no longer in use. This state can also be specified as the initial state by means of the FILSTAT parameter of the DFHFCT TYPE=FILE macro, or by defining a file using the RDO FILE options STATUS(UNENABLED) and OPENTIME(FIRSTREF).

The requested file is OPEN and in use by other transactions, but a CLOSE request against the file has been received.

This condition does not occur if the request is made to either a CLOSED, ENABLED file or a CLOSED, DISABLED file. In the first case, the file is opened as part of executing the request. In the second case, the DISABLED condition occurs.

It also cannot occur when the UNLOCK follows a successful READ for update or a WRITE MASSINSERT operation.

Default action: terminate the task abnormally.

SYSIDERR

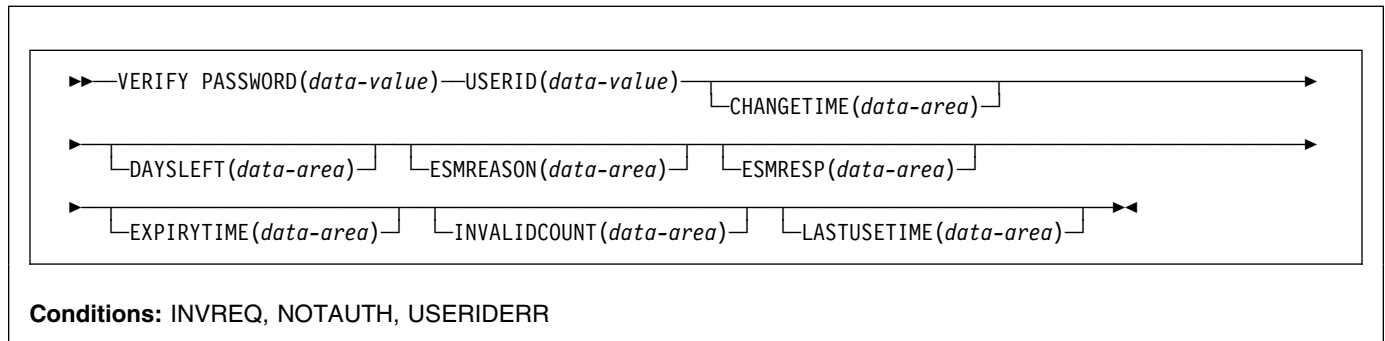
RESP2 values:

- 130** The SYSID option specifies a name that is neither the local system nor a remote system (made known to CICS by defining an RDO CONNECTION). SYSIDERR also occurs when the link to the remote system is closed.

Default action: terminate the task abnormally.

VERIFY PASSWORD

Allow an application to check that a password matches the password recorded by an external security manager (ESM).



Description

The VERIFY PASSWORD command allows an application to check that a password matches the password recorded by an external security manager (ESM) for a userid, and return values recorded by the external security manager for the password.

Unlike the SIGNON command, VERIFY PASSWORD does not depend upon the principal facility, so it can be issued when the facility is an APPC session.

If a VERIFY PASSWORD request is successful, you should not infer that a signon would also be successful. For example, the USERID may be revoked or may not be able to signon in this CICS region.

Attention: You should clear the password fields on the EXEC CICS commands that have a password option as soon as possible after use. This is to ensure that passwords are not revealed in system or transaction dumps.

Options

CHANGETIME(*data-area*)

returns the date and time the password was last changed, in ABSTIME units.

DAYSLEFT(*data-area*)

returns the number of days from now, in a halfword binary field, until the password expires. If the password is non-expiring, -1 is returned.

ESMREASON(*data-area*)

returns the reason code, in a fullword binary field, that CICS receives from the external security manager.

ESMRESP(*data-area*)

returns the response code, in a fullword binary field, that CICS receives from the external security manager.

EXPIRYTIME(*data-area*)

returns the date and time the password will expire, in ABSTIME units.

INVALIDCOUNT(*data-area*)

returns the number of times an invalid password was entered for this user.

LASTUSETIME(*data-area*)

returns the data and time this userid was last accessed, in ABSTIME units.

PASSWORD(*data-value*)

specifies the password, 8 characters, that you want the external security manager to check for the specified userid. The other data is not returned if the password is not valid.

USERID(*data-value*)

specifies the userid, 8 characters, of the user whose password is to be checked.

Note: In the CHANGETIME, LASTUSETIME, and EXPIRYTIME options, the time value returned is in the same format as the ASKTIME command, so it can be reformatted as a date and time, in a format specified by the caller, by using the FORMATTIME command.

Conditions

INVREQ

RESP2 values:

- 13** There is an unknown return code in ESMRESP from the external security manager.
- 18** The CICS external security manager interface is not initialized.
- 29** The external security manager is not responding.
- 32** The userid field contains a blank character in an invalid position.

VERIFY PASSWORD

Default action: terminate the task abnormally.

NOTAUTH

RESP2 values:

2 The supplied password is wrong.

3 A new password is required.

19 The USERID is revoked.

Default action: terminate the task abnormally.

USERIDERR

RESP2 values:

8 The USERID is not known to the external security manager.

Default action: terminate the task abnormally.

WAIT CONVID (APPC)

Ensure accumulated data is transmitted on an APPC mapped conversation.

```
▶▶—WAIT CONVID(name)—└─STATE(cvda)—◀◀
```

Conditions: INVREQ, NOTALLOC

Description

WAIT CONVID allows an application program to ensure that any accumulated application data and control indicators from a SEND command, or the results of a CONNECT PROCESS command, are transmitted to the partner transaction.

Options

CONVID(*name*)

identifies the conversation to which the command relates. The 4-character name identifies either the token returned by a previously executed ALLOCATE command in EIBRSRCE in the EIB, or the token representing the principal session (returned by a previously executed ASSIGN command).

STATE(*cvda*)

gets the state of the current conversation. The *cvda* values returned by CICS are:

```
ALLOCATED
CONFFREE
CONFRECEIVE
CONFSEND
FREE
PENDFREE
PENDRECEIVE
RECEIVE
ROLLBACK
SEND
SYNCFREE
SYNCRECEIVE
SYNCSEND
```

Conditions

INVREQ

RESP2 values:

200 Command not supported for distributed program link when it refers to the principal facility.

also occurs (RESP2 not set) in any of the following situations:

- The command is used on a conversation that is not using the EXEC CICS interface or that is not a mapped conversation.

Default action: terminate the task abnormally.

NOTALLOC

occurs if the CONVID value in the command does not relate to a conversation that is owned by the application.

Default action: terminate the task abnormally.

WAIT EVENT

WAIT EVENT

Wait for an event to occur.

```
▶▶—WAIT EVENT—ECADDR(ptr-value)—NAME(name)—◀◀
```

Condition: INVREQ

Note for transaction routing

Using this command could create inter-transaction affinities that adversely affect the use of dynamic transaction routing. See the *CICS Application Programming Guide* for more information about transaction affinities.

Description

WAIT EVENT synchronizes a task with the completion of an event initiated by the same task or by another task. The event would normally be the posting, at the expiration time, of a timer-event control area provided in response to a POST command, as described in “POST” on page 185. The WAIT EVENT command provides a method of directly relinquishing control to some other task until the event being waited on is completed.

CICS includes the addresses of all ECBs passed by WAIT EVENT commands of current tasks in the ECBLIST passed by CICS to the VSE WAIT facility when it runs out of work. Such ECBs can be posted using the VSE POST facility or by hand posting. Hand posting could, for example, be done by moving an appropriate value into the ECB.

A given ECB may not be waited on by more than one task at the same time. If this rule is not followed and the ECBLIST passed by CICS on the VSE WAIT contains duplicate ECB addresses, CICS abends.

Options

ECADDR(*ptr-value*)

specifies the address of the timer-event control area that must be posted before task activity can be resumed.

NAME(*name*)

specifies the symbolic name, 1–8 alphanumeric characters, that is returned in SUSPENDVALUE or HVALUE, when a task issues WAIT EVENT and is the subject of an INQUIRE TASK command or a CEMT INQ TASK.

Conditions

INVREQ

RESP2 values:

- 2 The ECB address is a null pointer, (X'00000000') or (X'FF000000').
- 3 The specified event control area address is above the 16MB line for programs executing in 24-bit mode.
- 4 The event control area address is not aligned on a fullword boundary.
- 6 The timer-event control area specified on a WAIT EVENT is in user-key task-lifetime storage, and is inaccessible to another transaction. This condition can only occur if the storage for the timer-event control area is obtained other than by a POST command, and is for posting as an ECB by some other task on completion of an event.

Note: CICS obtains storage for a timer-event control area in response to a POST command (and which can be used in conjunction with the WAIT EVENT command) from a shared subpool in user-key storage. This ensures that timer-event control areas are in shared storage and, when

referenced by a subsequent WAIT EVENT command, do not fail with an INVREQ.

Default action: terminate the task abnormally.

Examples

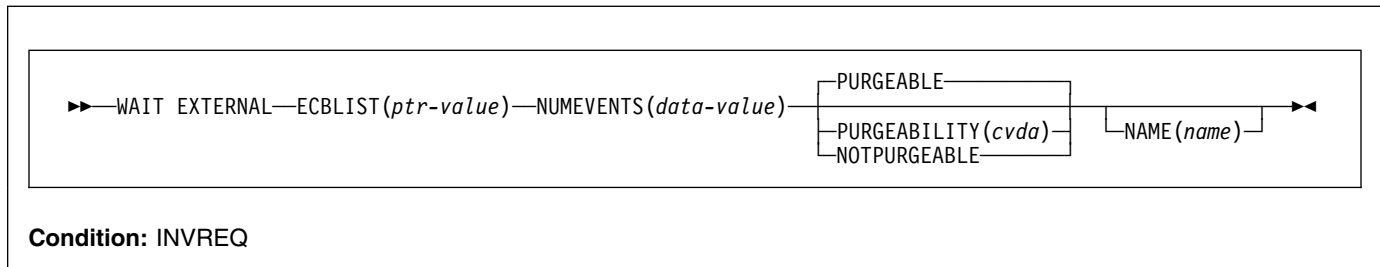
The following example shows you how to suspend the processing of a task until the specified event control area is posted:

```
EXEC CICS WAIT EVENT ECADDR(PVALUE)
```

WAIT EXTERNAL

WAIT EXTERNAL

Synchronize events.



Note for dynamic transaction routing

Using this command could create inter-transaction affinities that adversely affect the use of dynamic transaction routing. See the *CICS Application Programming Guide* for more information about transaction affinities.

Description

WAIT EXTERNAL waits for events that post VSE-format ECBs. The command causes the issuing task to be suspended until one of the ECBs has been posted, that is until one of the events has occurred. The task can wait on one or more ECBs. If it waits on more than one, it is dispatchable as soon as one of the ECBs is posted. You must ensure that each ECB is cleared (set to binary zeros) no later than the earliest time it could be posted. CICS cannot do this for you. If you wait on an ECB that has been previously posted and not subsequently cleared, your task is not suspended and continues to run as though the WAIT EXTERNAL had not been issued.

CICS includes the addresses of all ECBs passed by the WAIT EXTERNAL command of current tasks in the ECBLIST passed by CICS to the VSE WAIT facility when it runs out of work. Such ECBs can be posted using the VSE POST facility or by hand posting. Hand posting could be done, for example, by moving an appropriate value into the ECB. However, in CICS Transaction Server for OS/390®, any ECBs in a WAIT EXTERNAL ECBLIST are not available for hand posting, so to avoid future migration problems, you are recommended to use the WAIT EXTERNAL command only when you are sure that the ECBs will be posted by VSE POST. The WAITCICS command should be used if there is any possibility of an ECB being hand posted.

A given ECB must not be waited on by more than one task at the same time. If this rule is not followed, the second task to wait on the ECB gets an INVREQ condition.

Options

ECBLIST(*ptr-value*)

is a pointer to a list of addresses of VSE-format ECBs representing events. Both the ECBLIST and the ECBs can be above the 16MB line, that is they can be 31-bit addresses. Each ECB must be fullword aligned. Null (X'00000000' and X'FF000000') ECB addresses are ignored.

NAME(*name*)

specifies the symbolic name, 1–8 alphanumeric characters, that is returned in SUSPENDVALUE or HVALUE, when a task issues WAIT EXTERNAL and is the subject of an INQUIRE TASK command or a CEMT INQUIRE TASK.

NUMEVENTS(*data-value*)

is the number of such events, corresponding to the number of addresses in the ECBLIST. The field is fullword binary. When NUMEVENTS is specified as 1, ECBLIST must still be an address that points to a list containing just one ECB address.

PURGEABILITY(*cvda*)

determines the outcome of:

- An attempt to perform a deadlock time-out
- A SET TASK PURGEIFORCEPURGE command
- A CEMT SET TASK PURGEIFORCEPURGE

on the issuing task while it is waiting. The values passed to CICS are PURGEABLE (the default value), or NOTPURGEABLE. The outcome is:

Function	PURGEABLE	NOTPURGEABLE
DTIMOUT expired	Abend AEXY	No effect
CEMT SET TASK PURGE EXEC CICS SET TASK PURGE	Abend AEXY	No effect
CEMT SET TASK FORCEPURGE EXEC CICS SET TASK FORCEPURGE	Abend AEXY	Abend AEXY

See the *CICS Recovery and Restart Guide* for information about DTIMOUT and SET TASK PURGEIFORCEPURGE.

Conditions

INVREQ

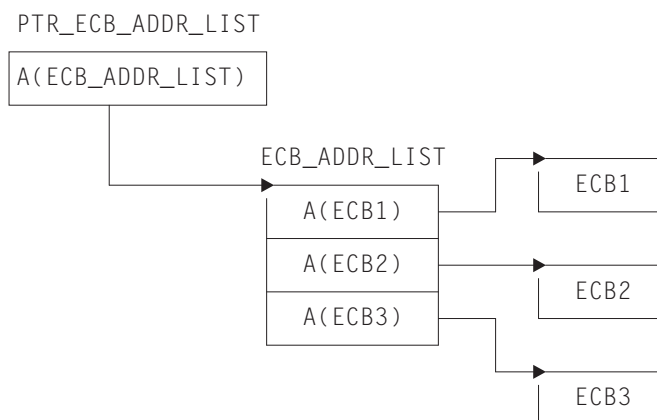
RESP2 values: CVDA values are:

- 1 An ECB is not valid, for example the ECB is not fullword aligned.
- 3 NUMEVENTS is not a positive number.
- 4 PURGEABILITY is specified with an incorrect CVDA.
- 5 No valid ECBs have been found in the list; either the ECBLIST address is not valid or all the ECB addresses are not valid.
- 6 Either:
 - The event control blocks (ECBs) specified are in user-key task-lifetime storage, and are inaccessible to another transaction that is expected to post the ECBs.
 - The ECBs specified are in read-only storage.

Default action: terminate the task abnormally.

Examples

The following figure shows how to use the ECBLIST parameter to point to a list of ECB addresses that in turn point to individual ECBs. Note that the ECBLIST variable is a pointer pointing to the first address of the list.



```

DCL
  ECB1      FIXED BIN(31),      /* actual ecb */
  ECB2      FIXED BIN(31),      /* actual ecb */
  ECB3      FIXED BIN(31);      /* actual ecb */
DCL          /* list of ecb addresses */
  1  ECB_ADDR_LIST,
  2  ECB_ADDR(3) PTR;
DCL          /* ptr to each addr list */
  PTR_ECB_ADDR_LIST PTR;
ECB_ADDR(1) = ADDR(ECB1);
ECB_ADDR(2) = ADDR(ECB2);
ECB_ADDR(3) = ADDR(ECB3);
/* set up pointer */
PTR_ECB_ADDR_LIST = ADDR(ECB_ADDR_LIST);
/* PTR_ECB_ADDR_LIST = ADDR(ECB_ADDR(1));
   (alternative) */

EXEC CICS WAIT EXTERNAL
      ECBLIST(PTR_ECB_ADDR_LIST)
      NUMEVENTS(3)
      PURGEABLE
  
```

Figure 2. ECBLIST option, EXEC CICS WAIT EXTERNAL

WAIT JOURNAL

WAIT JOURNAL

Synchronize with journal output.

```
▶▶—WAIT JOURNAL—JFILEID(data-value)—└─REQID(data-value)—└─STARTIO—▶▶
```

Conditions: INVREQ, IOERR, JIDERR, NOTOPEN

This command is now superseded by WAIT JOURNALNUM, but is still supported for compatibility with previous releases.

WAIT JOURNALNUM

Synchronize with journal output.

```

▶▶ WAIT JOURNALNUM(data-value) ┌─┐ ┌─┐ ┌─┐ ▶▶
                                └─┐ └─┐ └─┐
                                REQID(data-value) STARTIO
  
```

Conditions: INVREQ, IOERR, JIDERR, NOTOPEN

Description

WAIT JOURNALNUM synchronizes the task with the output of one or more journal records that have been created but whose output has been deferred; that is, with asynchronous journal output requests.

The journal records may already be written out from the journal buffer area to the corresponding system logger log stream, or the system logger output operation may be in progress. If the log stream output operation has already been completed, control returns immediately to the requesting task; if not, the requesting task waits until the operation has been completed. If STARTIO is specified, output is initiated immediately.

If the requesting program has made a succession of successful asynchronous output requests to the same journal, it is necessary to synchronize on only the last of these requests to ensure that all of the journal records have reached auxiliary storage. This may be done either by issuing a stand-alone WAIT JOURNALNUM command, or by making the last output command itself synchronous (by specifying the WAIT option in the WRITE JOURNALNUM command).

Options

JOURNALNUM(*data-value*)

Specifies the journal identifier as a halfword numeric value in the range 1–99. Journal numbers correspond to journal names as follows:

- Journal numbers in the range 2–99 refer to user journals named DFHJ02 through DFHJ99.
- Journal number 1 refers to the system log.

REQID(*data-value*)

specifies a fullword binary variable set to a number that refers to the journal record that has been created but possibly not yet written out.

If REQID is not specified, the task is synchronized with the output of the last record created for the journal specified by JOURNALNUM.

STARTIO

specifies that output of the journal record is to be initiated immediately.

Conditions

INVREQ

occurs if a WAIT JOURNALNUM command is issued before any WRITE JOURNALNUM command has been issued in the same task.

Default action: terminate the task abnormally.

IOERR

occurs if the physical output of a journal record was not accomplished because of an irrecoverable I/O error.

Default action: terminate the task abnormally.

JIDERR

occurs if the specified journal identifier does not exist in the journal control table (JCT).

Default action: terminate the task abnormally.

NOTOPEN

occurs if the WAIT JOURNALNUM command cannot be satisfied because the specified journal is not open.

Default action: terminate the task abnormally.

Examples

The following example shows how to request synchronization with the output of a journal record:

```

EXEC CICS WAIT JOURNALNUM(4)
      REQID(ENTRYID)
  
```

WAIT SIGNAL

WAIT SIGNAL

Suspend task on a logical unit.

▶▶—WAIT SIGNAL—◀◀

Conditions: NOTALLOC, SIGNAL, TERMERR

Description

WAIT SIGNAL, for a principal facility only, suspends a task until a SIGNAL condition occurs. Some logical units can interrupt the normal flow of data to the application program by a SIGNAL data-flow control command to CICS, signaling an attention, that in turn causes the SIGNAL condition to occur.

The HANDLE CONDITION SIGNAL command causes a branch to a user routine when an attention is received.

The logical units for which you can code a WAIT SIGNAL command are:

- LUTYPE4
- LUTYPE6.1
- 3600 (3601)
- 3767 interactive
- 3770 batch
- 3790 full-function

Conditions

NOTALLOC

occurs if the principal facility of the task is not a terminal.

Default action: terminate the task abnormally.

SIGNAL

occurs when the data-flow control command has been received from the principal facility.

EIBSIG is always set when an inbound signal is received.

Default action: ignore the condition.

TERMERR

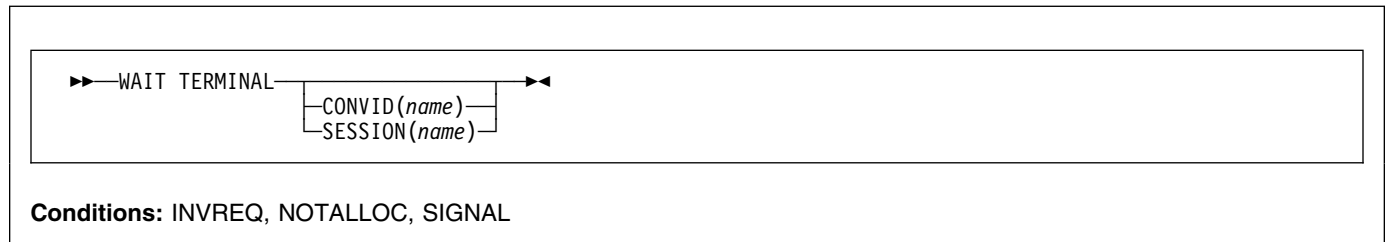
occurs for a terminal-related error.

A CANCEL TASK request by a user node error program (NEP) may cause this condition if the task has an outstanding terminal control request active when the node abnormal condition program (CSNE) handles the session error.

Default action: terminate the task abnormally withabend code ATNI.

WAIT TERMINAL

Ensure terminal operation has completed on an LUTYPE6.1 logical unit.



Description

WAIT TERMINAL ensures that terminal operation has completed.

Options

CONVID(*name*)

identifies the conversation to which the command relates. The 4-character name identifies either the token returned by a previously executed ALLOCATE command in EIBRSRCE in the EIB, or the token representing the principal session (returned by a previously executed ASSIGN command).

SESSION(*name*)

specifies the symbolic identifier (1–4 characters) of a session TCTTE. This option specifies the alternate facility to be used. If both this option and CONVID are omitted, the principal facility for the task is used.

Conditions

INVREQ

RESP2 values:

- 200** A distributed program link server application specified the function-shipping session (its principal facility) on the CONVID option.

Default action: terminate the task abnormally.

NOTALLOC

occurs if the facility specified in the command is not owned by the application.

Default action: terminate the task abnormally.

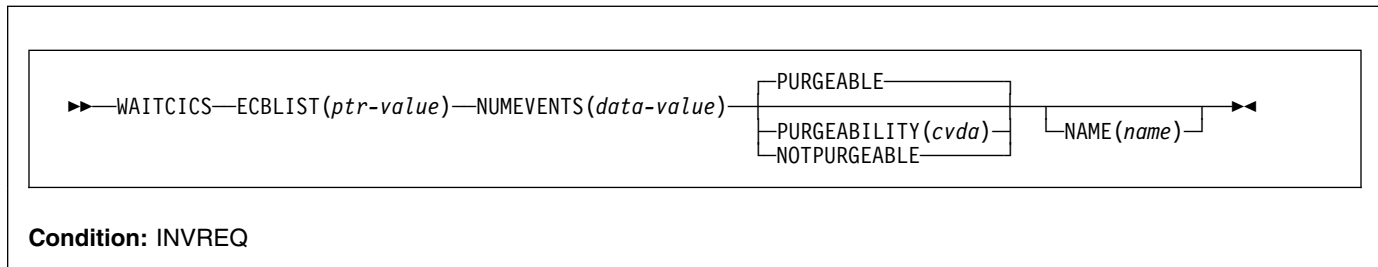
SIGNAL

occurs when an inbound SIGNAL data-flow control command is received from a logical unit or session. EIBSIG is always set when an inbound signal is received.

Default action: ignore the condition.

WAITCICS

Synchronize events.



Condition: INVREQ

Note for dynamic transaction routing

Using this command could create inter-transaction affinities that adversely affect the use of dynamic transaction routing. See the *CICS Application Programming Guide* for more information about transaction affinities.

Description

WAITCICS waits for events that post VSE-format ECBs. The command causes the issuing task to be suspended until one of the ECBs has been posted, that is until one of the events has occurred. The task can wait on one or more ECBs. If it waits on more than one, it is dispatchable as soon as one of the ECBs is posted. You must ensure that each ECB is cleared, set to binary zeros, no later than the earliest time it could be posted. CICS cannot do this for you. If you wait on an ECB that has been previously posted and not subsequently cleared, your task is not suspended and continues to run as though the WAITCICS had not been issued.

CICS includes the addresses of all ECBs passed by WAITCICS commands of current tasks in the ECBLIST passed by CICS to the VSE WAIT facility when it runs out of work. Such ECBs can be posted using the VSE POST facility or by hand posting. Hand posting could, for example, be done by moving an appropriate value into the ECB. If hand posting is definitely not going to be used, it is preferable to use WAIT EXTERNAL.

A given ECB may not be waited on by more than one task at the same time. If this rule is not followed and the ECBLIST passed by CICS on the VSE WAIT contains duplicate ECB addresses, VSE abends CICS.

Options

ECBLIST(ptr-value)

is a pointer to a list of addresses of VSE-format ECBs representing events. Both the ECBLIST and the ECBs can be above the 16MB line, that is they can be 31-bit addresses. Each ECB must be fullword aligned. Null

(X'00000000' and X'FF000000') ECB addresses are ignored.

NAME(name)

specifies the symbolic name, 1–8 alphanumeric characters, as the reason for the wait. The value you specify is returned in the SUSPENDVALUE or HVALUE option respectively of the EXEC CICS INQUIRE TASK or CEMT INQUIRE TASK commands.

NUMEVENTS(data-value)

is the number of such events, corresponding to the number of addresses in the ECBLIST. The field is fullword binary. When NUMEVENTS is specified as one, ECBLIST must still be an address that points to a list containing just one ECB address.

PURGEABILITY(cvda)

causes the issuing task to be suspended until one of the ECBs has been posted; that is, until one of the events has occurred. The values passed to CICS are PURGEABLE (the default value), or NOTPURGEABLE. The field is fullword binary. If while this task is waiting another function attempts to purge it, the result is as follows:

Function	PURGEABLE	NOTPURGEABLE
DTIMOUT expired	Abend AEXY	No effect
CEMT SET TASK PURGE EXEC CICS SET TASK PURGE	Abend AEXY	No effect
CEMT SET TASK FORCEPURGE EXEC CICS SET TASK FORCEPURGE	Abend AEXY	Abend AEXY

See the *CICS Recovery and Restart Guide* for information about DTIMOUT and the *CICS System*

Programming Reference manual for more information about SET TASK.

Conditions

INVREQ

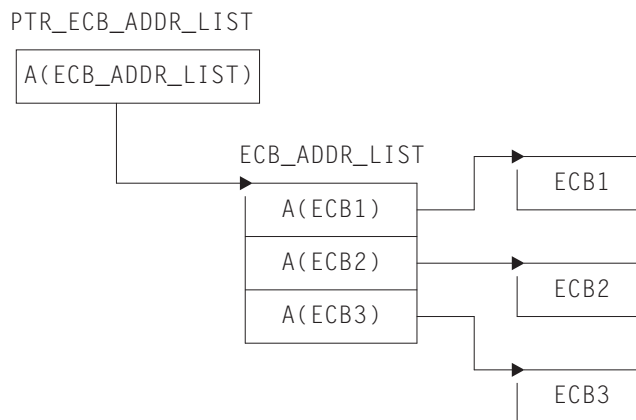
RESP2 values:

- 1 An ECB is not valid, for example the ECB is not fullword aligned.
- 3 NUMEVENTS is not a positive number.
- 4 PURGEABILITY is specified with an incorrect CVDA.
- 5 No valid ECBs have been found in the list; either the ECBLIST address is not valid, or all the ECB addresses are not valid.
- 6 Either:
 - The event control blocks (ECBs) specified are in user-key task-lifetime storage, and are inaccessible to another transaction that is expected to post the ECBs.
 - The ECBs specified are in read-only storage.

Default action: terminate the task abnormally.

Examples

The following figure shows how to use the ECBLIST parameter to point to a list of ECB addresses that in turn point to individual ECBs. Note that the ECBLIST variable is a pointer pointing to the first address of the list.



```

DCL
  ECB1      FIXED BIN(31),      /* actual ecb */
  ECB2      FIXED BIN(31),      /* actual ecb */
  ECB3      FIXED BIN(31);      /* actual ecb */
DCL
  /* list of ecb addresses */
  1 ECB_ADDR_LIST,
  2 ECB_ADDR(3) PTR;
DCL
  /* ptr to each addr list */
  PTR_ECB_ADDR_LIST PTR;
ECB_ADDR(1) = ADDR(ECB1);
ECB_ADDR(2) = ADDR(ECB2);
ECB_ADDR(3) = ADDR(ECB3);
/* set up pointer */
PTR_ECB_ADDR_LIST = ADDR(ECB_ADDR_LIST);
/* PTR_ECB_ADDR_LIST = ADDR(ECB_ADDR(1));
   (alternative) */

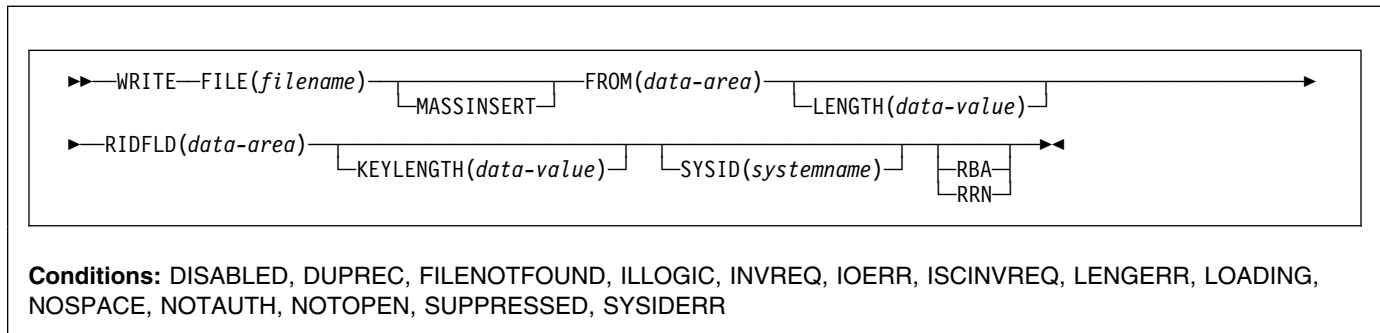
EXEC CICS WAITCICS
      ECBLIST(PTR_ECB_ADDR_LIST)
      NUMEVENTS(3)
      PURGEABLE
  
```

Figure 3. ECBLIST option, EXEC CICS WAITCICS

WRITE

WRITE

Write a record.



Description

WRITE writes a new record to a file on a local or a remote system.

When this command is used to write a record to a CICS-maintained data table, the update is made to both the source VSAM KSDS and the in-memory data table, unless the XDTAD user exit rejects the record from the table. The details of the command for a CICS-maintained table are the same as for a VSAM KSDS.

When this command is used to write a record to a user-maintained data table, the update is made to the in-memory data table (unless rejected by the XDTAD user exit).

For a VSAM ESDS, the record is always added at the end of the data set. CICS does not use the identification field specified in RIDFLD when calculating the RBA of the new record, but the new RBA is returned to the application in the record identification field specified in the RIDFLD option.

For a VSAM KSDS, the record is added in the location specified by the associated key; this location may be anywhere in the data set. For VSAM data sets, the key in the record and the key in the RIDFLD identification field must be the same.

For a VSAM ESDS or KSDS, records can be either fixed-length or variable-length. MASSINSERT operations must proceed with ascending keys, and must be terminated by an UNLOCK before any other request to the same data set.

Options

FILE(filename)

specifies the name of the file to be accessed.

If SYSID is specified, the data set to which this file refers is assumed to be on a remote system irrespective of whether the name is defined in the FCT. Otherwise, the FCT entry is used to find out whether the data set is on a local or a remote system.

FROM(data-area)

specifies the record that is to be written to the data set referred to by this file.

KEYLENGTH(data-value)

specifies the length (halfword binary) of the key that has been specified in the RIDFLD option, except when RBA or RRN is specified, in which case KEYLENGTH is not valid. You must code KEYLENGTH if you are also using SYSID (unless you are also using RBA or RRN). If the length specified is different from the length defined for the data set, the INVREQ condition occurs.

LENGTH(data-value)

specifies the length, as a halfword binary value, of the record to be written.

You must specify this option if you are using SYSID.

You must also specify the option for a file with variable-length records (including user-maintained data tables). It need not be specified for fixed-length records, but its inclusion is recommended because it causes a check to be made that the record being written is not longer than that defined for the data set.

MASSINSERT

(VSAM) specifies that the WRITE command is part of a mass-insert operation, that is, a series of WRITES each specifying MASSINSERT.

You cannot use MASSINSERT for user-maintained data tables.

RBA

(VSAM ESDS base data sets only) specifies that the record identification field specified in the RIDFLD option contains a relative byte address. Use this option only when writing to an ESDS base.

RIDFLD(*data-area*)

specifies the record identification field. The contents can be a key, a relative byte address, or relative record number (for VSAM data sets), or a block reference, a physical key, and a deblocking argument (for DAM data sets). For a relative byte address or a relative record number, the format of this field must be fullword binary. If RBA is specified, RIDFLD contains the relative byte address (greater than or equal to zero) of the record to be written. If RRN is specified, RIDFLD contains the relative record number (greater than or equal to 1) of the record to be written.

When adding records to a keyed data set, the field must contain the complete key.

RRN

(VSAM RRDS) specifies that the record identification field specified in the RIDFLD option contains a relative record number.

SYSID(*systemname*)

specifies the name of the system to which the request is directed.

If you specify SYSID, and omit both RBA and RRN, you must also specify LENGTH and KEYLENGTH; they cannot be found in the FCT.

LENGTH must either be specified explicitly or must be capable of being generated by the compiler from the FROM option. See "LENGTH options" on page 5 for more information about specifying LENGTH.

Conditions

Note: RESP2 values are not set for files that are on remote systems.

DISABLED

RESP2 values:

50 A file is disabled because it was initially defined as disabled and has not since been enabled.

A file has been disabled by a SET FILE or a CEMT SET FILE command.

Default action: terminate the task abnormally.

DUPREC

RESP2 values:

150 An attempt is made to add a record to a data set by referring to a file, or a path over a file (with the UNIQUEKEY attribute), in which the same key already exists.

Default action: terminate the task abnormally.

FILENOTFOUND

RESP2 values:

1 A file name referred to in the FILE option cannot be found in the FCT.

Default action: terminate the task abnormally.

ILLOGIC

RESP2 values:

110 A VSAM error occurs that does not fall within one of the other CICS response categories.

(See EIBRCODE in the EXEC interface block; refer to Appendix A, "EXEC interface block" on page 373 for details.)

Default action: terminate the task abnormally.

INVREQ

RESP2 values:

20 Add operations are not allowed according to the file entry specification in the FCT.

23 When writing records containing embedded keys, the key in the record area (FROM option) and the key in RIDFLD do not match.

26 The KEYLENGTH option is specified, and the specified length does not equal the length defined for the data set that this file refers to.

38 A WRITE with the MASSINSERT option is issued against a DAM file.

40 A DAM key conversion error occurred.

44 The WRITE command does not conform to the format of WRITE for a user-maintained data table (for example, MASSINSERT or RBA is specified).

49 An attempt was made to write to a file that is an invalid file type.

Default action: terminate the task abnormally.

IOERR

RESP2 values:

120 There is an I/O error during the file control operation. An I/O error is any unusual event that is not covered by a CICS condition. (Further information is available in the EXEC interface block; refer to Appendix A, "EXEC interface block" on page 373 for details.)

For VSAM files, IOERR normally indicates a hardware error.

For DAM files, IOERR could mean that you are trying to write to a DAM track address that is not defined for the data set.

Default action: terminate the task abnormally.

ISCINVREQ

RESP2 values:

WRITE

- 70** The remote system indicates a failure that does not correspond to a known condition.

Default action: terminate the task abnormally.

LENGERR

RESP2 values:

- 12** The length specified for the write operation exceeds the maximum record size; the record is truncated.
- 10** LENGTH is omitted for a WRITE to a file with variable-length records or to a DAM file with undefined format records.
- 14** An incorrect length is specified for a write operation involving fixed-length records.

LOADING

RESP2 values:

- 104** A WRITE is issued to a user-maintained data table that is currently being loaded. A user-maintained data table cannot be modified during loading.

Default action: terminate the task abnormally.

NOSPACE

RESP2 values:

- 100** No space is available on the direct access device for adding records to a data set.
- 102** The maximum number of table entries specified for the user-maintained table has already been reached.
- 103** Either:
- CICS is unable to get sufficient storage in the file-owning region's data space to create an in-memory table entry for the record being written.
 - CICS is unable to get sufficient storage in the CICS address space for its own internal control blocks.

Default action: terminate the task abnormally.

NOTAUTH

RESP2 values:

- 101** A resource security check has failed on FILE(filename).

Default action: terminate the task abnormally.

NOTOPEN

RESP2 values:

- 60** The requested file is CLOSED and UNENABLED. The CLOSED, UNENABLED state is reached after a CLOSE request has been received against an OPEN ENABLED file and the file is no longer in use. This state can also be specified as the initial state by means of the FILSTAT parameter of the DFHFCT TYPE=FILE macro, or by defining a file

using the RDO FILE options STATUS(UNENABLED) and OPENTIME(FIRSTREF).

The requested file is OPEN and in use by other transactions, but a CLOSE request against the file has been received.

This condition does not occur if the request is made to either a CLOSED, ENABLED file or a CLOSED, DISABLED file. In the first case, the file is opened as part of executing the request. In the second case, the DISABLED condition occurs.

Default action: terminate the task abnormally.

SUPPRESSED

RESP2 values:

- 105** A user exit program that is invoked at the XDTAD exit point decides not to add the record to the user-maintained data table.

Default action: terminate the task abnormally.

SYSIDERR

RESP2 values:

- 130** The SYSID option specifies a name that is neither the local system nor a remote system (made known to CICS by defining a CONNECTION). SYSIDERR also occurs when the link to the remote system is closed.

Default action: terminate the task abnormally.

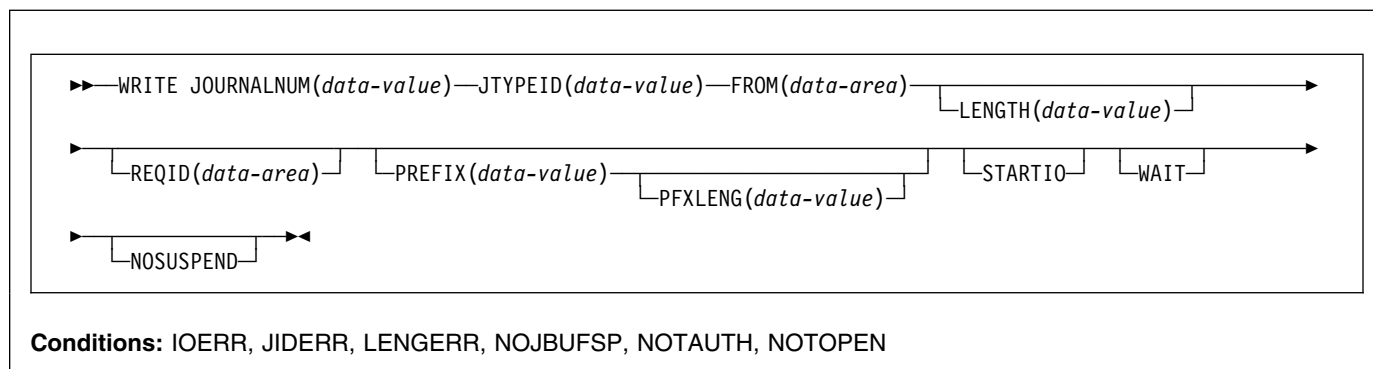
Examples

For example:

```
EXEC CICS WRITE
      FROM(RECORD)
      LENGTH(DATLEN)
      FILE('MASTER')
      RIDFLD(KEYFLD)
```


WRITE JOURNALNUM

Create a journal record.



Description

WRITE JOURNALNUM creates a journal record. The request can be for synchronous or asynchronous output; definitions of these terms, and detailed information regarding the synchronization of journal output, are contained in the *CICS Application Programming Guide*.

Options

FROM(data-area)

specifies the user data to be built into the journal record.

JOURNALNUM(data-value)

Specifies the journal identifier as a halfword numeric value in the range 1–99. Journal numbers correspond to journal names as follows:

- Journal numbers in the range 2–99 refer to user journals named DFHJ02 through DFHJ99.
- Journal number 1 refers to the system log.

JTYPEID(data-value)

specifies a 2-character identifier to be placed in the journal record to identify its origin.

LENGTH(data-value)

specifies the length (halfword binary value) in bytes of the user data to be built into the journal record. The minimum value is 0 and the maximum value is (buffer size–76) minus PFXLENG.

NOSUSPEND

specifies that an application program is not to be suspended for the NOJBUFSP condition.

PFXLENG(data-value)

specifies the length (halfword binary value) in bytes of the user prefix data to be included in the journal record. The minimum value is 0 and the maximum value is (buffer size–72) minus LENGTH.

PREFIX(data-value)

specifies the user prefix data to be included in the journal record. A data area must be provided in COBOL programs.

REQID(data-area)

specifies a data area that identifies the journal record. The data area is a fullword binary variable. CICS sets the variable to a number that depends upon the position in the data set of the record being created. REQID applies to asynchronous output (WAIT option not specified).

STARTIO

specifies that output of the journal record is to be initiated immediately. If WAIT is specified for a journal with a low utilization, STARTIO should be specified also to prevent the requesting task waiting for the journal buffer to be filled. Very high utilization ensures that the buffer is cleared quickly, so that STARTIO is unnecessary.

WAIT

specifies that synchronous journal output is required. The requesting task waits until the record has been written.

WRITE JOURNALNUM

Conditions

IOERR

occurs if the physical output of a journal record was not accomplished because of an irrecoverable I/O error.

Default action: terminate the task abnormally.

JIDERR

occurs if the specified journal identifier does not exist in the journal control table (JCT).

Default action: terminate the task abnormally.

LENGERR

occurs if the computed length for the journal record exceeds the total buffer space allocated for the journal, as specified in the JCT entry for the journal, or if the length specified for the prefix or for the data is negative.

Default action: terminate the task abnormally.

NOJBUFSP

occurs if the journal buffers are logically full (that is, the current buffer is full, and I/O is in progress on the alternate buffer) or the remaining space in the current buffer is insufficient for this journal record.

Default action: suspend task activity until the journal request can be satisfied. CICS ensures that both buffers are written out to auxiliary storage, thus freeing them for new records. (The default can be overridden by the NOSUSPEND option.)

NOTAUTH

occurs when a resource security check has failed on JOURNALNUM(data-value).

Default action: terminate the task abnormally.

NOTOPEN

occurs if the journal command cannot be satisfied because the specified journal is not open.

Default action: terminate the task abnormally.

Examples

The following example shows how to request synchronous journal output and wait for the output operation to be completed:

```
EXEC CICS WRITE
  JOURNALNUM(2)
  JTYPEID('XX')
  FROM(KEYDATA)
  LENGTH(8)
  PREFIX(PROGNAME)
  PFXLENG(6)
  WAIT
```

In this example, STARTIO is not specified, so the task waits until the journal buffer is full or until output is initiated by a

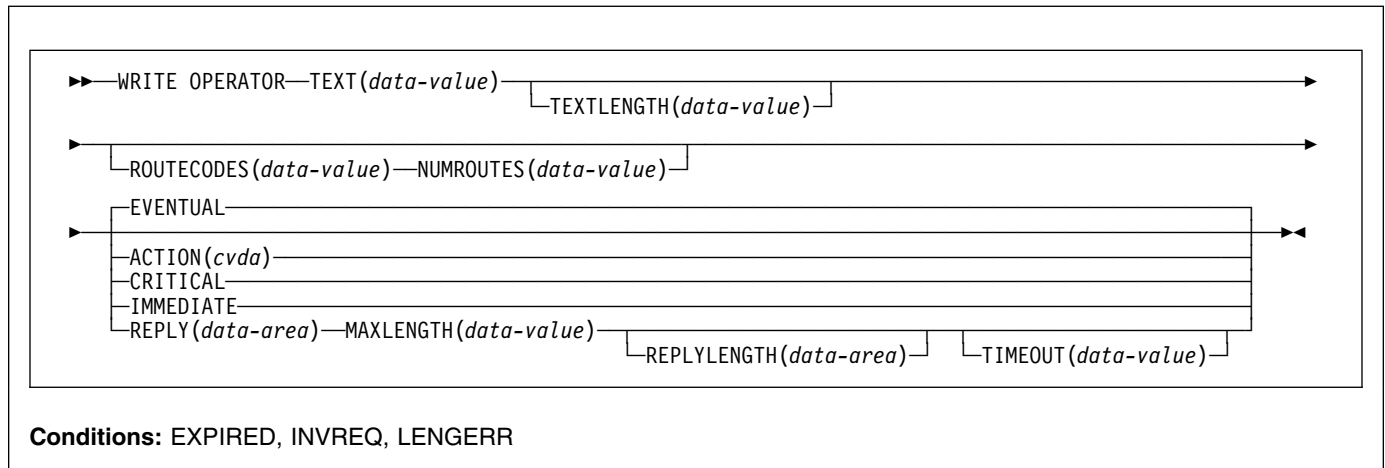
STARTIO request in another task. CICS limits the wait to one second.

The following example shows how to request deferred (asynchronous) journal output:

```
EXEC CICS WRITE
  JOURNALNUM(1)
  JTYPEID('SD')
  FROM(COMDATA)
  LENGTH(10)
  REQID(ENTRYID)
```

WRITE OPERATOR

Write a message on the system console.



Description

WRITE OPERATOR enables an application to write a message to one or more system consoles and, if necessary, wait for a reply. The command can specify route codes. This is of particular use to application packages that need to issue their own operator messages.

As a result of a change in the way CICS handles messages sent to the console, text lengths of greater than 113 characters are split into two lines.

None of the variables below can be defined as PL/I variable character strings.

Note: If ACTION (or one of the equivalent CVDA values below) is specified, the message is retained until the console operator explicitly deletes it or CICS terminates.

The action code is identical with the descriptor code to be associated with the message. Only one of the descriptor codes 2, 3, or 11 may be specified for this parameter.

If ACTION is not specified, no descriptor code is associated with the message. The descriptor codes have the following meanings:

- 2** Immediate action
- 3** Eventual action
- 11** Critical eventual action.

The CRITICAL option is equivalent to a specification of ACTION(11). The EVENTUAL option is equivalent to a specification of ACTION(3). The IMMEDIATE option is equivalent to a specification of ACTION(2). Refer to your system programmer for information about how this command affects the appearance of the console screen to the operator.

Options

ACTION(*cvda*)

specifies an action code to be associated with this message. CVDA values are:

CRITICAL

specifies that the message requires eventual action by the operator and has enough critical importance to remain on the console screen. The message remains on the screen until it is deleted by the operator.

EVENTUAL

specifies that the operator should take action when there is time. The message is rolled off when other messages fill up the screen.

IMMEDIATE

specifies that the operator should take action immediately. The message remains on the console screen until it is deleted by the operator.

MAXLENGTH(*data-value*)

is a fullword binary field that contains the length of the reply area (in the range 1–119 bytes). You must specify MAXLENGTH if you specify REPLY.

NUMROUTES(*data-value*)

is a fullword binary field that defines the number of routing codes.

REPLY(*data-area*)

provides a data area for receiving the operator's reply. If you specify this option, your application pauses until either a reply is received or the TIMEOUT period expires.

WRITE OPERATOR

REPLYLENGTH(*data-area*)

specifies the actual length (fullword binary value) of the operator's reply.

ROUTECODES(*data-value*)

is a variable-length field. Each code is one byte and contains a binary number in the range 1–28. The default is a single code, set to 2. In COBOL programs only, you must use a data-area that contains the 1-byte values rather than a data-value.

TEXT(*data-value*)

is a data value containing the text to be sent. In COBOL programs only, you must use a data-area that contains the text to be sent to the operator, rather than a data-value.

TEXTLENGTH(*data-value*)

specifies the length (fullword binary value) of the text.

If the REPLY option is specified, the length is in the range 0–121 bytes.

If the REPLY option is not specified, the length is in the range 0–690 bytes.

If the length of the text is greater than 113, CICS formats the message in a multiline write to operator (WTO); each line has 69 bytes with a maximum of ten lines.

The output is edited in such a way that each line is broken, if possible, at a space character. The next line starts with a non-space character. If there is no room to reformat the data within the overall limit of 690 bytes of ten lines of 69 bytes, the output is not reformatted.

TIMEOUT(*data-value*)

is a fullword binary field that contains the maximum time (in seconds) that CICS waits for a reply before returning control to this transaction. This must be in the range 0–86 400 (24 hours). The system default value is specified by the OPERTIM system initialization parameter. You can only specify TIMEOUT if you have also specified REPLY.

Conditions

EXPIRED

RESP2 values:

- 7 TIMEOUT has occurred before the operator's reply was received.

Default action: terminate the task abnormally.

INVREQ

RESP2 values:

- 1 The TEXTLENGTH value is not valid.
- 2 The NUMROUTES value is not valid.
- 3 The ROUTECODES value is not valid.
- 4 The MAXLENGTH value is not valid.

- 5 The TIMEOUT value is not valid.

- 6 The ACTION value is not valid.

Default action: terminate the task abnormally.

LENGERR

RESP2 values:

- 8 The reply was longer than MAXLENGTH, and has been truncated.

Default action: terminate the task abnormally.

WRITEQ TD

Write data to transient data queue.

```

▶▶ WRITEQ TD—QUEUE(name)—FROM(data-area)—LENGTH(data-value)—SYSID(systemname)—▶▶

```

Conditions: DISABLED, INVREQ, IOERR, ISCVREQ, LENGERR, NOSPACE, NOTAUTH, NOTOPEN, QIDERR, SYSIDERR

Description

WRITEQ TD writes transient data to a predefined symbolic destination.

Options

FROM(*data-area*)

specifies the data that is to be written to the transient data queue.

LENGTH(*data-value*)

specifies the length (halfword binary value) of the data to be written.

QUEUE(*name*)

specifies the symbolic name (1–4 alphanumeric characters) of the queue to be written to. The named queue must have been defined to CICS.

SYSID(*systemname*)

(remote systems only) specifies the name (1–4 characters) of the system to which the request is directed.

If SYSID is specified, the data set is assumed to be on a remote system irrespective of whether or not the name is defined as remote in the DCT. Otherwise the entry in the DCT is used to find out whether the data set is on a local or a remote system.

Conditions

DISABLED

occurs when the queue has been disabled.
Default action: terminate the task abnormally.

INVREQ

occurs if WRITEQ names an extrapartition queue that has been opened for input.

Note: This condition cannot be raised for intrapartition queues.

Default action: terminate the task abnormally.

IOERR

occurs when an input/output error occurs and the data record in error is skipped.

Default action: terminate the task abnormally.

ISCVREQ

occurs when the remote system indicates a failure that does not correspond to a known condition.

Default action: terminate the task abnormally.

LENGERR

occurs in any of the following situations:

- WRITEQ names an extrapartition queue and does not specify a length consistent with the RECFORM and RECSIZE options specified in the DFHDCT TYPE=SDSCI macro. that defined the queue. The check is made after the XTDOUT exit has been invoked; this exit may change the length of the data to be passed to the access method.
- WRITEQ names an intrapartition queue and does not specify a length consistent with the control interval defined for the intrapartition data set. Again, the check is made after the XTDOUT exit has been invoked.

Default action: terminate the task abnormally.

NOSPACE

occurs if no more space exists on the intrapartition or extrapartition queue, or the relative byte address (RBA) for an intrapartition queue would exceed 2 gigabytes. When this happens, no more data should be written to the queue because it may be lost.

Default action: terminate the task abnormally.

NOTAUTH

occurs when a resource security check has failed on QUEUE(*name*).

Default action: terminate the task abnormally.

WRITEQ TD

NOTOPEN

occurs if the destination is closed.

Note: This condition cannot be raised for intrapartition queues.

Default action: terminate the task abnormally.

QIDERR

occurs if the symbolic destination to be used with a transient data control command cannot be found.

Default action: terminate the task abnormally.

SYSIDERR

occurs when the SYSID option specifies a name that is neither the local system nor a remote system (made known to CICS by defining a CONNECTION).

SYSIDERR also occurs when the link to the remote system is closed.

Default action: terminate the task abnormally.

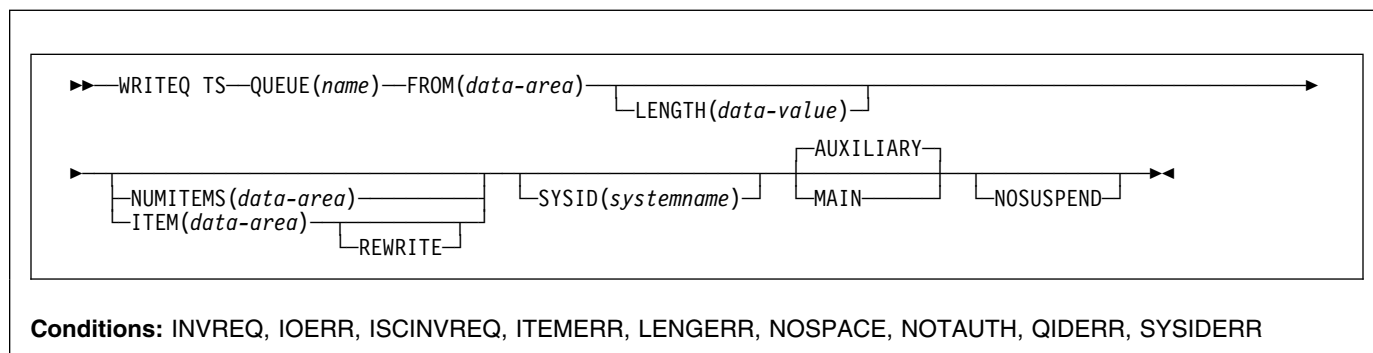
Examples

The following example shows how to write data to a predefined symbolic destination; in this case, the control system message log (CSML):

```
EXEC CICS WRITEQ TD
      QUEUE('CSML')
      FROM(MESSAGE)
      LENGTH(LENG)
```

WRITEQ TS

Write data to a temporary storage queue.



Note for dynamic transaction routing

Using this command could create inter-transaction affinities that adversely affect the use of dynamic transaction routing. See the *CICS Application Programming Guide* for more information about transaction affinities.

Description

WRITEQ TS stores temporary data (records) in a temporary storage queue in main or auxiliary storage.

If a queue has been defined as recoverable, the program must not issue a WRITEQ TS if a DELETEQ TS has previously been issued within the same logical unit of work. In other words, following a DELETEQ TS, no WRITEQ TS can be issued until after a syncpoint has occurred.

If there is insufficient space available in the temporary storage data set to satisfy the WRITEQ TS request, the task is suspended until space does become available. (Space may be released by other tasks in the system.) If, however, space is not available in the temporary storage data set, and the NOSUSPEND option has been specified, or there is an active HANDLE CONDITION for NOSPACE, the NOSPACE condition is raised.

Options

AUXILIARY

specifies that the temporary storage queue is on a direct access storage device in auxiliary storage. This option is ignored for an existing queue.

This is the default value for the first write.

FROM(data-area)

specifies the data to be written to temporary storage.

ITEM(data-area)

specifies, as a halfword binary value, the item number of the logical record to be replaced in the queue (REWRITE option also specified).

ITEM can be both an input and output field to CICS. As such, programmers should ensure that the ITEM field is not defined within protected storage when issuing a WRITEQ command. If the ITEM value were a literal (for example), command checking (CMDPROT=YES system initialization parameter) would result in an AEYD abend occurring.

Note: In earlier releases, ITEM on a WRITEQ TS without REWRITE would perform a similar function to NUMITEMS. This function is retained for compatibility.

LENGTH(data-value)

specifies the length, as a halfword binary value, of the data to be written.

You must specify this option if you are using SYSID.

For a description of a safe upper limit, see "LENGTH options" on page 5.

MAIN

specifies that the temporary storage queue is in main storage. This option is ignored for an existing queue.

If you use the MAIN option to write data to a temporary storage queue on a remote system, the data is stored in main storage if the remote system is accessed by the CICS multiregion operation (MRO) facility. If these conditions are not met, the data is stored in auxiliary storage.

If the system is MRO and MAIN is specified, the queue is not recoverable and SYNCPOINT ROLLBACK does not function.

NOSUSPEND

specifies that the application program is not to be suspended if there is insufficient space in the temporary

WRITEQ TS

storage data set to satisfy the WRITEQ TS request. Instead, the NOSPACE condition is raised.

Note, however, that if a HANDLE CONDITION for NOSPACE is active when the command is executed, this also overrides the default action, and control is passed to the user label supplied in the HANDLE CONDITION. This takes precedence over the NOSUSPEND option but is, of course, negated by either NOHANDLE or RESP.

This does not apply to temporary storage queues in main storage (MAIN option).

NUMITEMS(*data-area*)

specifies a halfword binary field into which CICS stores a number indicating how many items there are now in the queue, after the WRITEQ TS command is executed.

If the record starts a new queue, the item number assigned is 1; subsequent item numbers follow on sequentially. NUMITEMS is not valid if REWRITE is specified.

QUEUE(*name*)

specifies the symbolic name (1-8 characters) of the queue to be written to. If the queue name appears in the TST, and the entry is marked as remote, the request is shipped to a remote system. The name must be unique within the CICS system. Do not use X'FA' through X'FF', or **, or \$\$, or DF, as the first character of the name; these characters are reserved for CICS use. The name cannot consist solely of binary zeros.

REWRITE

specifies that the existing record in the queue is to be overwritten with the data provided. If the REWRITE option is specified, the ITEM option must also be specified. If the specified queue does not exist, the QIDERR condition occurs. If the correct item within an existing queue cannot be found, the ITEMERR condition occurs and the data is not stored.

SYSID(*systemname*)

(remote queues only) specifies the system name (1–4 characters) identifying the remote system to which the request is to be sent.

Conditions

INVREQ

occurs in any of the following situations:

- A WRITEQ TS command specifies a queue name that consists solely of binary zeros.
- A WRITEQ TS command specifies a queue that is locked and awaiting ISC session recovery.
- The queue was created by CICS internal code.

Default action: terminate the task abnormally.

IOERR

occurs when there is an irrecoverable input/output error.

Default action: terminate the task abnormally.

ISCINVREQ

occurs when the remote system indicates a failure that does not correspond to a known condition.

Default action: terminate the task abnormally.

ITEMERR

occurs in any of the following situations:

- The item number specified in a WRITEQ TS command with the REWRITE option, is not valid (that is, it is outside the range of entry numbers assigned for the queue).
- The maximum number of items (32 767) is exceeded.

Default action: terminate the task abnormally.

LENGERR

occurs if the length of the stored data is zero or negative.

Default action: terminate the task abnormally.

NOSPACE

occurs when the NOSUSPEND option is specified and there is no space for the data in:

- The auxiliary temporary storage data set
- The temporary storage pool list structure

Also occurs if there is no space and there is an active HANDLE CONDITION for NOSPACE.

Default action: ignore the condition.

NOTAUTH

occurs when a resource security check has failed on QUEUE(*name*).

Default action: terminate the task abnormally.

QIDERR

occurs when the queue specified by a WRITEQ TS command with the REWRITE option cannot be found, either in:

- Main storage
- Auxiliary storage

Default action: terminate the task abnormally.

SYSIDERR

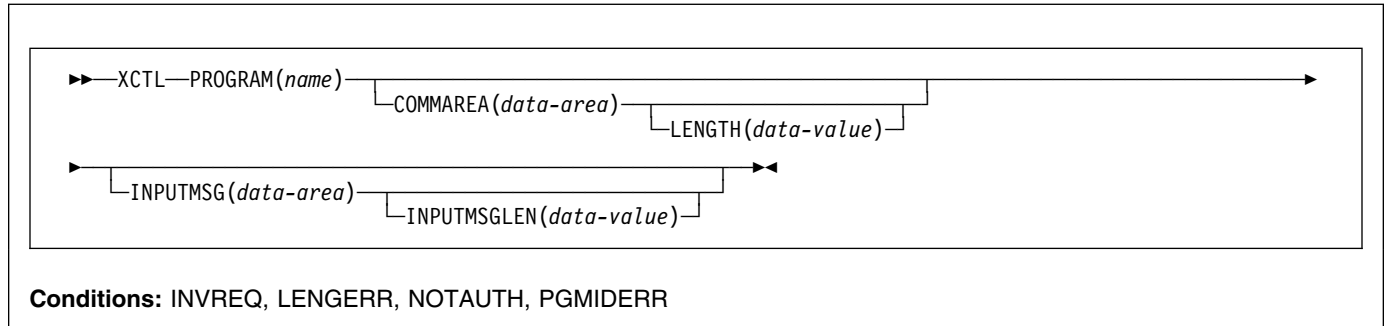
occurs in any of the following situations:

- When the SYSID option specifies a name that is neither the local system nor a remote system (made known to CICS by defining a CONNECTION).
- The link to the remote system is closed.

Default action: terminate the task abnormally.

XCTL

Transfer program control.



Description

XCTL transfers control from one application program to another at the same logical level. The program from which control is transferred is released. If the program to which control is transferred is not already in main storage, it is loaded.

Options

COMMAREA(*data-area*)

specifies a communication area to be made available to the invoked program. In this option the contents of the *data-area* are passed. In COBOL, you must give this *data-area* the name DFHCOMMAREA in the receiving program. (See the section about passing data to other programs in the *CICS Application Programming Guide*.)

INPUTMSG(*data-area*)

specifies data to be passed to the invoked program when it first issues a RECEIVE command. If the invoked program passes control to another program by a LINK command, a linked chain is created, as described under the INPUTMSG option of the LINK command. The INPUTMSG data remains available until a RECEIVE command is issued or until control returns to CICS.

INPUTMSGLEN(*data-value*)

specifies a halfword binary value that specifies the length of the data passed by INPUTMSG.

LENGTH(*data-value*)

specifies the length (halfword binary data value) in bytes of the communication area. For a description of a safe upper limit, see "LENGTH options" on page 5.

PROGRAM(*name*)

specifies the identifier (1–8 alphanumeric characters) of the program to which control is to be passed unconditionally. The specified name must have been defined as a program to CICS, though if AUTOINSTALL is active a definition is autoinstalled.

Conditions

INVREQ

RESP2 values:

- 8** An XCTL command with the INPUTMSG option is issued for a program that is not associated with a terminal, or that is associated with an APPC logical unit, or an IRC session.
- 29** EXEC XCTL is not allowed in a GLUE or TRUE.
- 30** The program manager domain has not yet been initialized. This is probably due to a XCTL request having been made in a first stage PLT.
- 200** An XCTL command with the INPUTMSG option is issued in a program invoked by DPL.

Default action: terminate the task abnormally.

LENGERR

RESP2 values:

- 11** LENGTH is less than 0 or greater than 32763.
- 26** The COMMAREA address passed was zero, but LENGTH was non-zero.
- 27** INPUTMSGLEN was less than 0 or greater than 32767.
- 28** LENGTH or INPUTMSGLEN is greater than the length of the data area specified in the COMMAREA or INPUTMSG options, and while that data was being copied a destructive overlap occurred because of the incorrect length.

Default action: terminate the task abnormally.

NOTAUTH

RESP2 values:

- 101** A resource security check has failed on PROGRAM(*name*).

XCTL

Default action: terminate the task abnormally.

PGMIDERR

RESP2 values:

- 1 A program has no entry in the PPT and either program autoinstall was switched off, or the program autoinstall control program indicated that the program should not be autoinstalled.
- 2 The program is disabled.
- 3 A program could not be loaded because:
 - This was the first load of the program and the program load failed, usually because the load module could not be found.
 - This was a subsequent load of the program, but the first load failed.

In order to reset the load status the load module must be in a sublibrary of the LIBDEF search chain for the CICS startup jobstream, and a SET PROGRAM NEWCOPY will be required

- 9 The installed program definition is for a remote program.
- 21 The program autoinstall control program failed either because the program autoinstall control program is incorrect, incorrectly defined, or as a result of an abend in the program autoinstall control program. Program autoinstall is disabled and message DFHPG0202 or DFHPG0203 written to the CSPL.
- 22 The model returned by the program autoinstall control program was not defined in the PPT table, or was not enabled.
- 23 The program autoinstall control program returned invalid data.
- 24 Define for the program failed due to autoinstall returning an invalid program name or definition.

Default action: terminate the task abnormally.

Examples

The following example shows how to request a transfer of control to an application program called PROG2:

```
EXEC CICS XCTL PROGRAM('PROG2')
```

Appendix A. EXEC interface block

This appendix contains a description of the EXEC interface block (EIB). An application program can read all the fields in the EIB of the associated task by name, but must not change the contents of any of them other than through an EXEC CICS command.

For each field, the contents and format (for each of the application programming languages COBOL, C, PL/I, and ASM) are given. All fields contain zeros in the absence of meaningful information. Fields are listed in alphabetical order.

EIB fields

EIBAID

contains the attention identifier (AID) associated with the last terminal control or basic mapping support (BMS) input operation from a display device such as the 3270.

```
COBOL: PIC X(1).
C:      unsigned char eibaid;
PL/I:   CHAR(1)
ASM:    CL1
```

EIBATT

indicates that the RU contains attach header data (X'FF').

```
COBOL: PIC X(1).
C:      unsigned char eibatt;
PL/I:   CHAR(1)
ASM:    CL1
```

EIBCALEN

contains the length of the communication area that has been passed to the application program from the last program, using the COMMAREA and LENGTH options. If no communication area is passed, this field contains zeros.

```
COBOL: PIC S9(4) COMP.
C:      short int eibcalen;
PL/I:   FIXED BIN(15)
ASM:    H
```

EIBCOMPL

indicates, on a terminal control RECEIVE command, whether the data is complete (X'FF'). If the NOTRUNCATE option has been used on the RECEIVE command, CICS retains data in excess of the amount requested via the LENGTH or MAXLENGTH option. EIBRECV is set indicating that further RECEIVE

commands are required. EIBCOMPL is not set until the last of the data has been retrieved.

EIBCOMPL is always set when a RECEIVE command without the NOTRUNCATE option is executed.

```
COBOL: PIC X(1).
C:      unsigned char eibcompl;
PL/I:   CHAR(1)
ASM:    CL1
```

EIBCONF

indicates that a CONFIRM request has been received (X'FF') for an APPC conversation.

```
COBOL: PIC X(1).
C:      unsigned char eibconf;
PL/I:   CHAR(1)
ASM:    CL1
```

EIBCPASN

contains the cursor address (position) associated with the last terminal control or basic mapping support (BMS) input operation from a display device such as the 3270.

```
COBOL: PIC S9(4) COMP.
C:      short int eibcpasn;
PL/I:   FIXED BIN(15)
ASM:    H
```

EIBDATE

contains the date the task is started; this field is updated by the ASKTIME command. The date is in packed decimal form (0CYYDDD+) where C shows the century with values 0 for the 1900s and 1 for the 2000s. For example, the dates 31 December 1999 and 1 January 2000 have EIBDATE values of 0099365 and 0100001 respectively.

APAR PQ87874

added the next sentence

EIBDATE is updated from the COMREG JOBDATE field, which can be initially set using the VSE // DATE statement in the CICS startup JCL.

```
COBOL: PIC S9(7) COMP-3.
C:      char eibdate [4];
PL/I:   FIXED DEC(7,0)
ASM:    PL4
```

EIBDS • EIBFN

EIBDS

contains the symbolic identifier of the last data set referred to in a file control request.

```
COBOL: PIC X(8).  
C:      char eibds [8];  
PL/I:   CHAR(8)  
ASM:    CL8
```

EIBEOC

indicates that an end-of-chain indicator appears in the RU just received (X'FF').

```
COBOL: PIC X(1).  
C:      unsigned char eibeoc;  
PL/I:   CHAR(1)  
ASM:    CL1
```

EIBERR

indicates that an error has been received (X'FF') on an APPC conversation.

```
COBOL: PIC X(1).  
C:      unsigned char eiberr;  
PL/I:   CHAR(1)  
ASM:    CL1
```

EIBERRCD

when EIBERR is set, contains the error code that has been received. The following values can be returned in the first two bytes of EIBERRCD:

X'0889' Conversation error detected.
X'0824' SYNCPOINT ROLLBACK requested.

```
COBOL: PIC X(4).  
C:      char eiberrcd [4];  
PL/I:   CHAR(4)  
ASM:    CL4
```

See the *CICS Distributed Transaction Programming Guide* for information about other EIBERRCD values that can occur.

EIBFMH

indicates that the user data just received or retrieved contains an FMH (X'FF').

```
COBOL: PIC X(1).  
C:      unsigned char eibfmh;  
PL/I:   CHAR(1)  
ASM:    CL1
```

EIBFN

contains a code that identifies the last CICS command issued by the task.

```
COBOL: PIC X(2).  
C:      char eibfn [2];  
PL/I:   CHAR(2)  
ASM:    CL2
```

Code Command

0202 ADDRESS
0204 HANDLE CONDITION
0206 HANDLE AID
0208 ASSIGN
020A IGNORE CONDITION
020C PUSH HANDLE
020E POP HANDLE
0210 ADDRESS SET

0402 RECEIVE
0404 SEND
0406 CONVERSE
0408 ISSUE EODS
040A ISSUE COPY
040C WAIT TERMINAL
040E ISSUE LOAD
0410 WAIT SIGNAL
0412 ISSUE RESET
0414 ISSUE DISCONNECT
0416 ISSUE ENDOUTPUT
0418 ISSUE ERASEAUP
041A ISSUE ENDFILE
041C ISSUE PRINT
041E ISSUE SIGNAL
0420 ALLOCATE
0422 FREE
0424 POINT
0426 BUILD ATTACH
0428 EXTRACT ATTACH
042A EXTRACT TCT
042C WAIT CONVID
042E EXTRACT PROCESS
0430 ISSUE ABEND
0432 CONNECT PROCESS
0434 ISSUE CONFIRMATION
0436 ISSUE ERROR
0438 ISSUE PREPARE
043A ISSUE PASS
043C EXTRACT LOGONMSG
043E EXTRACT ATTRIBUTES

0602 READ
0604 WRITE
0606 REWRITE
0608 DELETE
060A UNLOCK
060C STARTBR
060E READNEXT
0610 READPREV
0612 ENDBR

0614 RESETBR

0802 WRITEQ TD
0804 READQ TD
0806 DELETEQ TD0A02 WRITEQ TS
0A04 READQ TS
0A06 DELETEQ TS0C02 GETMAIN
0C04 FREEMAIN0E02 LINK
0E04 XCTL
0E06 LOAD
0E08 RETURN
0E0A RELEASE
0E0C ABEND
0E0E HANDLE ABEND1002 ASKTIME
1004 DELAY
1006 POST
1008 START
100A RETRIEVE
100C CANCEL1202 WAIT EVENT
1204 ENQ
1206 DEQ
1208 SUSPEND1402 WRITE JOURNALNUM
1404 WAIT JOURNALNUM

1602 SYNCPOINT

1802 RECEIVE MAP
1804 SEND MAP
1806 SEND TEXT
1808 SEND PAGE
180A PURGE MESSAGE
180C ROUTE
180E RECEIVE PARTN
1810 SEND PARTNSET
1812 SEND CONTROL

1C02 DUMP

1E02 ISSUE ADD
1E04 ISSUE ERASE
1E06 ISSUE REPLACE
1E08 ISSUE ABORT
1E0A ISSUE QUERY
1E0C ISSUE END
1E0E ISSUE RECEIVE
1E10 ISSUE NOTE
1E12 ISSUE WAIT
1E14 ISSUE SEND

2002 BIF DEEDIT

4802 ENTER TRACENUM
4804 MONITOR4A02 ASKTIME ABSTIME
4A04 FORMATTIME
5602 SPOOLOPEN
5604 SPOOLREAD
5606 SPOOLWRITE
5610 SPOOLCLOSE
5620 SPOOLOPEN REPORT
5622 SPOOLOPEN MAPNAME
5624 SPOOLOPEN ESCAPE
5626 SPOOLOPEN RESUME
5628 SPOOLWRITE REPORT
562A SPOOLWRITE MAPNAME
562C SPOOLCLOSE REPORT5E06 CHANGE TASK
5E22 WAIT EXTERNAL
5E32 WAITCICS6A02 QUERY SECURITY
6C02 WRITE OPERATOR
6C12 CICSMESSAGE7402 SIGNON
7404 SIGNOFF
7406 VERIFY PASSWORD
7408 CHANGE PASSWORD

7E02 DUMP TRANSACTION

Codes 82nn and 84nn are used by the CICS Front End Programming Interface. See the *CICS Front End Programming Interface User's Guide* for a description.

EIBFREE

indicates that the application program cannot continue using the facility. The application program should either free the facility or should terminate so that the facility is freed by CICS (X'FF').

```
COBOL: PIC X(1).
C:      unsigned char eibfree;
PL/I:   CHAR(1)
ASM:    CL1
```

EIBNODAT

indicates that no data has been sent by the remote application (X'FF'). A message has been received from the remote system that conveyed only control information. For example, if the remote application executed a SEND command with the WAIT option, any data would be sent across the link. If the remote application then executed a SEND INVITE command without using the FROM option to transmit data at the

EIBRCODE

same time, it would be necessary to send the INVITE instruction across the link by itself. In this case, the receiving application finds EIBNODAT set. The use of this field is restricted to application programs holding conversations across APPC links only.

```
COBOL: PIC X(1).
C:      unsigned char eibnodat;
PL/I:   CHAR(1)
ASM:    CL1
```

EIBRCODE

contains the CICS response code returned after the function requested by the last CICS command to be issued by the task has been completed.

Note: For new commands where EIBRESP and EIBRESP2 are the strategic means of interrogating the resulting condition of an executed command, byte 3 of EIBRCODE has the same value as EIBRESP. Any further information is in EIBRESP2 rather than EIBRCODE. For a normal response, this field contains hexadecimal zeros (6 X'00').

Almost all of the information in this field can be used within application programs by the HANDLE CONDITION command.

```
COBOL: PIC X(6).
C:      char eibrancode [6];
PL/I:   CHAR(6)
ASM:    CL6
```

The following list contains the values of the bytes together with the names of the conditions associated with the return codes.

See the notes at the end of the list of values for explanations of the numbers following some of the conditions.

EIBFN	EIBRCODE	Condition
02 ..	E0	INVREQ
04 ..	04	EOF
04 ..	10	EODS
04 ..	C1	EOF
04 ..	C2	ENDINPT
04 ..	D0	SYSIDERR (see note 1)
04 ..	D2	SESSIONERR (see note 2)
04 ..	D3	SYSBUSY (see note 3)
04 ..	D4	SESSBUSY
04 ..	D5	NOTALLOC
04 ..	E0	INVREQ (see note 4)
04 ..	E1	LENGERR (see note 5)
04 ..	E3	WRBRK
04 ..	E4	RDATT
04 ..	E5	SIGNAL
04 ..	E6	TERMIDERR
04 ..	E7	NOPASSBKRD
04 ..	E8	NOPASSBKWR
04 ..	EA	IGREQCD

EIBFN	EIBRCODE	Condition
04 ..	EB	CBIDERR
04 ..	EC	PARTNERIDERR
04 ..	ED	NETNAMEIDERR
04 ..	F1	TERMERR
04 20	EOC
04 40	INBFMH
04 F6	NOSTART
04 F7	NONVAL
06 ..	01	FILENOTFOUND
06 ..	02	ILLOGIC (see note 6)
06 ..	08	INVREQ
06 ..	0C	NOTOPEN
06 ..	0D	DISABLED
06 ..	0F	ENDFILE
06 ..	80	IOERR (see note 6)
06 ..	81	NOTFND
06 ..	82	DUPREC
06 ..	83	NOSPACE
06 ..	84	DUPKEY
06 ..	85	SUPPRESSED
06 ..	86	LOADING
06 ..	D0	SYSIDERR (see note 1)
06 ..	D1	ISCINVREQ
06 ..	D6	NOTAUTH
06 ..	E1	LENGERR
08 ..	01	QZERO
08 ..	02	QIDERR
08 ..	04	IOERR
08 ..	08	NOTOPEN
08 ..	10	NOSPACE
08 ..	C0	QBUSY
08 ..	D0	SYSIDERR (see note 1)
08 ..	D1	ISCINVREQ
08 ..	D6	NOTAUTH
08 ..	D7	DISABLED
08 ..	E0	INVREQ
08 ..	E1	LENGERR
0A ..	01	ITEMERR
0A ..	02	QIDERR
0A ..	04	IOERR
0A ..	08	NOSPACE
0A ..	20	INVREQ
0A ..	D0	SYSIDERR (see note 1)
0A ..	D1	ISCINVREQ
0A ..	D6	NOTAUTH
0A ..	E1	LENGERR
0C ..	E1	LENGERR
0C ..	E2	NOSTG
0E ..	01	PGMIDERR
0E ..	D6	NOTAUTH
0E ..	E0	INVREQ
0E ..	E1	LENGERR
10 ..	01	ENDDATA
10 ..	04	IOERR
10 ..	11	TRANSIDERR
10 ..	12	TERMIDERR
10 ..	20	EXPIRED
10 ..	81	NOTFND
10 ..	D0	SYSIDERR (see note 1)
10 ..	D1	ISCINVREQ

EIBFN	EIBRCODE	Condition
10 ..	D6	NOTAUTH
10 ..	D8	USERIDERR
10 ..	E1	LENGERR
10 ..	E9	ENVDEFERR
10 ..	FF	INVREQ
12 ..	32	ENQBUSY
12 ..	E0	INVREQ
12 ..	E1	LENGERR
14 ..	01	JIDERR
14 ..	02	INVREQ
14 ..	05	NOTOPEN
14 ..	06	LENGERR
14 ..	07	IOERR
14 ..	09	NOJBUFSP
14 ..	D6	NOTAUTH
16 ..	01	ROLLEDBACK
18 ..	01	INVREQ
18 ..	02	RETPAGE
18 ..	04	MAPFAIL
18 ..	08	INVMPSZ (see note 7)
18 ..	20	INVERRTERM
18 ..	40	RTESOME
18 ..	80	RTEFAIL
18 ..	E1	LENGERR
18 ..	E3	WRBRK
18 ..	E4	RDATT
18 02	PARTNFAIL
18 04	INVPARTN
18 08	INVPARTNSET
18 10	INVLDC
18 20	UNEXPIN
18 40	IGREQCD
18 80	TSIOERR
18 01	OVERFLOW
18 04	EODS
18 08	EOC
18 10	IGREQID
1A ..	E0	INVREQ
1A ..	04	DSSTAT
1A ..	08	FUNCERR
1A ..	0C	SELNERR
1A ..	10	UNEXPIN
1A ..	E1	LENGERR
1A 11	EODS
1A 2B	IGREQCD
1A 20	EOC
22 ..	80	INVEXITREQ
4A 01	INVREQ
56 0D	NOTFND
56 10	INVREQ
56 13	NOTOPEN
56 14	ENDFILE
56 15	ILLOGIC
56 16	LENGERR
56 2A	NOSTG
56 46	NOTAUTH
56 50	NOSPOOL
56 55	ALLOCERR
56 56	STRELERR

EIBFN	EIBRCODE	Condition
56 57	OPENERR
56 58	SPOLBUSY
56 59	SPOLERR
56 5A	NODEIDERR

Notes:

1. When SYSIDERR occurs, further information is provided in bytes 1 and 2 of EIBRCODE, as follows:

```

.. 04 00 .. . . . . request was for a function
                        that is not valid.
.. 04 04 .. . . . . no session available and
                        NOQUEUE.
.. 04 08 .. . . . . modename not found2.
.. 04 0C .. . . . . modename not valid2.
.. 04 10 .. . . . . task canceled or timed
                        out during allocation2.
.. 04 14 .. . . . . mode group is out of
                        service2.
.. 04 18 .. . . . . close - DRAIN=ALL2.
.. 08 .. . . . . sysid is not available.
.. 08 00 .. . . . . no session available,
                        all sessions are out
                        of service, or released,
                        or being quiesced.
.. 08 04 .. . . . . no session available,
                        request to queue rejected
                        by XZIQUE global user
                        exit program.
.. 08 08 .. . . . . no session available;
                        request rejected by XZIQUE
                        global user exit program.
.. 0C xx .. . . . . sysid definition error.
.. 0C 00 .. . . . . name not that of TCTSE.
.. 0C 04 .. . . . . name not that of remote
                        TCTSE.
.. 0C 08 .. . . . . mode name not found.
.. 0C 0C .. . . . . profile not found

```

Further information about SYSIDERR can be found in the

2. When SESSIONERR occurs, further information is provided in bytes 1 and 2 of EIBRCODE, as follows:

```

.. 08 .. . . . . session out of service
.. 0C xx .. . . . . session definition error
.. 0C 00 .. . . . . name not found
.. 0C 0C .. . . . . profile not found.

```

3. If SYSBUSY occurs on an ALLOCATE command that attempts to acquire a session to an APPC terminal or system, byte 3 of the EIBRCODE indicates where the error condition was detected:

EIBRECV

```
.. . . . 00 .. . . the request was for a
                session to a connected
                terminal or system.
.. . . . 01 .. . . the request was for a
                session to a remotely
                connected terminal or
                system, and the error
                occurred in the terminal-
                owning region (TOR) or
                an intermediate system.
.. . . . 02 .. . . the request was for a
                session to a remotely
                connected terminal or
                system, and the error
                occurred in the
                application-owning
                region (AOR).
```

Further information about SYSBUSY can be found in the

4. When INVREQ occurs during terminal control operations, further information is provided in bytes 1 or 3 of EIBRCODE as follows:

```
.. 24 .. . . . . ISSUE PREPARE command -
                STATE error.
.. . . . 04 .. . . ALLOCATE command - TCTTE
                already allocated.
.. . . . 08 .. . . FREE command - TCTTE in
                wrong state.
.. . . . 0C .. . . CONNECT PROCESS command -
                SYNCLVL 2 requested, but
                cannot be supported on
                the session in use.
.. . . . 10 .. . . EXTRACT ATTACH command -
                incorrect data.
.. . . . 14 .. . . SEND command - CONFIRM
                option specified, but
                conversation not SYNCLVL 1.
.. . . . 18 .. . . EXTRACT TCT command -
                incorrect netname.
.. . . . 1C .. . . an incorrect command has
                been issued for the terminal
                or logical unit in use.
.. . . . 20 .. . . an incorrect command has
                been issued for the LUTYPE6.2
                conversation type in use.
.. . . . 28 .. . . GETMAIN failure on ISSUE
                PASS command.
.. . . . 2C .. . . Command invalid in DPL
                environment.
```

5. When LENGERR occurs during terminal control operations, further information is provided in byte 1 of EIBRCODE, as follows:

```
.. 00 .. . . . . input data is overlong and
                has been truncated.
.. 04 .. . . . . on output commands, an
                incorrect (FROM)LENGTH has
                been specified, either less
                than zero or greater than
                32 767.
.. 08 .. . . . . on input commands, an
                incorrect (TO)LENGTH has
                been specified, greater than
                32 767.
.. 0C .. . . . . length error has occurred on
                ISSUE PASS command.
```

Note: This field is not used exclusively for the above and may take other values.

6. When ILLOGIC or IOERR occurs during file control operations, further information is provided in field EIBRCODE, as follows:

```
.. xx xx .. . . . . DAM response (ERRBYTE value)
.. xx .. . . . . . . VSAM return code.
.. . . xx .. . . . . VSAM error code.
```

See the *VSE/ESA Messages and Codes Volume 2* for VSAM return codes, and the *VSE/ESA System Macro Users Guide* for DAM ERRBYTE values.

7. When INVMPSZ occurs during BMS operations, byte 3 of field EIBRCODE contains the terminal code:

```
.. . . . xx .. . . terminal code.
```

These are the same as the mapset suffixes shown in Table 31 on page 431.

8. Codes 82nn and 84nn are used by the CICS Front End Programming Interface. See the *CICS Front End Programming Interface User's Guide* for a description.

EIBRECV

indicates that the application program is to continue receiving data from the facility by executing RECEIVE commands (X'FF').

```
COBOL: PIC X(1).
C:      unsigned char eibrecv;
PL/I:   CHAR(1)
ASM:    CL1
```

² For APPC only.

EIBREQID

contains the request identifier assigned to an interval control command by CICS; this field is not used when a request identifier is specified in the application program.

```
COBOL: PIC X(8).
C: char eibreqid [8];
PL/I: CHAR(8)
ASM: CL8
```

EIBRESP

contains a number corresponding to the RESP condition that occurred. These numbers are listed below (in decimal) for the conditions that can occur during execution of the commands described in this manual.

```
COBOL: PIC S9(8) COMP
C: long int eibresp;
PL/I: FIXED BIN(31)
ASM: F
```

No.	Condition	No.	Condition
00	NORMAL	47	SELNERR
01	ERROR	48	FUNCERR
02	RDATT	49	UNEXPIN
03	WRBRK	50	NOPASSBKRD
04	EOF	51	NOPASSBKWR
05	EODS	53	SYSIDERR
06	EOC	54	ISCINVREQ
07	INBFMH	55	ENQBUSY
08	ENDINPT	56	ENVDEFERR
09	NONVAL	57	IGREQCD
10	NOSTART	58	SESSIONERR
11	TERMIDERR	59	SYSBUSY
12	FILENOTFOUND	60	SESSBUSY
13	NOTFND	61	NOTALLOC
14	DUPREC	62	CBIDERR
15	DUPKEY	63	INVEXITREQ
16	INVREQ	64	INVPARTNSET
17	IOERR	65	INVPARTN
18	NOSPACE	66	PARTNFAIL
19	NOTOPEN	69	USERIDERR
20	ENDFILE	70	NOTAUTH
21	ILLOGIC	72	SUPPRESSED
22	LENGERR	73	WRONGSTAT
23	QZERO	74	NAMEERROR
24	SIGNAL	75	RESIDERR
25	QBUSY	76	CCERROR
26	ITEMERR	77	MAPERROR
27	PGMIDERR	80	NOSPOOL
28	TRANSIDERR	81	TERMERR
29	ENDDATA	82	ROLLEDBACK
31	EXPIRED	84	DISABLED
32	RETPAGE	85	ALLOCERR
33	RTEFAIL	86	STRELERR
34	RTESOME	87	OPENERR
35	TSIOERR	88	SPOLBUSY
36	MAPFAIL	89	SPOLEERR
37	INVERRTERM	90	NODEIDERR
38	INVMPSTZ	91	TASKIDERR
39	IGREQID	92	TCIDERR
40	OVERFLOW	93	DSNNOTFOUND

No.	Condition	No.	Condition
41	INVLDC	94	LOADING
42	NOSTG	95	MODELIDERR
43	JIDERR	96	OUTDESCERR
44	QIDERR	97	PARTNERIDERR
45	NOJBUFSP	98	PROFILEIDERR
46	DSSTAT	99	NETNAMEIDERR

EIBRESP2

contains more detailed information that may help explain why the RESP condition occurred. This field contains meaningful values, as documented with each command to which it applies. For requests to remote files, EIBRESP2 contains zeros.

For programs written in C, any value passed via the *exit* or *return* function is saved in EIBRESP2.

```
COBOL: PIC S9(8) COMP.
C: long int eibresp2;
PL/I: FIXED BIN(31)
ASM: F
```

EIBRLDBK

indicates rollback.

```
COBOL: PIC X(1).
C: unsigned char eibrldb;
PL/I: CHAR(1)
ASM: CL1
```

EIBRSRCE

contains the symbolic identifier of the resource being accessed by the latest executed command:

Type of command	Resource	Number of characters (H=halfword binary)
BMS	Map name	7
File control	File name	7
Interval control	Transaction name	4
Journal control	Journal number	H
Program control	Program name	8
Temporary storage control	TS queue name	8
Terminal control	Terminal or logical unit identifier	4
	or LU6.1 session or APPC conversation identifier	4
Transient data control	TD queue name	4

Identifiers less than eight characters in length are padded on the right with blanks.

EIBSIG • EIBTRNID

```
COBOL: PIC X(8).  
C:      char eibrsrc [8];  
PL/I:   CHAR(8)  
ASM:    CL8
```

```
COBOL: PIC S9(7) COMP-3.  
C:      char eibtime [4];  
PL/I:   FIXED DEC(7,0)  
ASM:    PL4
```

EIBSIG

indicates that SIGNAL has been received (X'FF').

```
COBOL: PIC X(1).  
C:      unsigned char eibsig;  
PL/I:   CHAR(1)  
ASM:    CL1
```

EIBTRMID

contains the symbolic terminal identifier of the principal facility (terminal or logical unit) associated with the task.

```
COBOL: PIC X(4).  
C:      char eibtrmid [4];  
PL/I:   CHAR(4)  
ASM:    CL4
```

EIBSYNC

indicates that the application program must take a syncpoint or terminate. Before either is done, the application program must ensure that any other facilities, owned by it, are put into the send state, or are freed (X'FF').

```
COBOL: PIC X(1).  
C:      unsigned char eibsync;  
PL/I:   CHAR(1)  
ASM:    CL1
```

EIBTRNID

contains the symbolic transaction identifier of the task.

```
COBOL: PIC X(4).  
C:      char eibtrnid [4];  
PL/I:   CHAR(4)  
ASM:    CL4
```

EIBSYNRB

indicates that the application program should issue a SYNCPOINT ROLLBACK command (X'FF'). This field is only set in application programs holding a conversation on an APPC or MRO link.

```
COBOL: PIC X(1).  
C:      unsigned char eibsynrb;  
PL/I:   CHAR(1)  
ASM:    CL1
```

EIBTASKN

contains the task number assigned to the task by CICS. This number appears in trace table entries generated while the task is in control. The format of the field is packed decimal.

```
COBOL: PIC S9(7) COMP-3.  
C:      char eibtasn [4];  
PL/I:   FIXED DEC(7,0)  
ASM:    PL4
```

EIBTIME

contains the time at which the task is started (this field is updated by the ASKTIME command). The time is in packed decimal form (0HHMMSS+).

Appendix B. Codes returned by ASSIGN

This appendix describes the codes returned by the ASSIGN command.

ASSIGN TERMCODE

This section gives the meanings of the terminal type codes in the first byte of the data area returned by the TERMCODE option of the ASSIGN command.

The codes are derived from the DEVICE attribute of the RDO TYPETERM resource definition. The second byte of the data area contains a model number in character form, as set by the TERMMODEL attribute of the RDO TYPETERM resource definition.. TYPETERM is described in the *CICS Resource Definition Guide*.

The codes are listed here as both bit patterns and hexadecimal values.

Code	Meaning
.... ...1	X'01' 7770
.... ..1.	X'02' System 7
.... 1...	X'08' Console
...1 ..1.	X'12' Sequential disk
...1 .1..	X'14' Magnetic tape
...1 1...	X'18' Card reader or line printer
...1 1..1	X'19' Spooling system printer
...1 1.1.	X'1A' Spooling internal reader
..1.	X'20' Hard-copy terminals
..1. ...1	X'21' Model 33/35 TWX
..1. .1.	X'22' Teletypewriter
..1. .1..	X'24' 1050
..1. 1...	X'28' 2740
..1. 1.1.	X'2A' 2741 Correspondence
..1. 1.11	X'2B' 2741 EBCDIC
.1..	X'40' Video terminals
.1.. ...1	X'41' 2260 local
.1.. 1...	X'48' 2260 remote
.1.. 1.1.	X'4A' 1053
.1.. 11..	X'4C' 2265
1...	X'80' Bisynchronous
1... ..1.	X'82' 2770
1... .1..	X'84' 2780
1... .1.1	X'85' 3780
1... ..11.	X'86' 2980
1... 1...	X'88' 3735
1... 1..1	X'89' 3740
1... 1.1.	X'8A' 3600 bisynchronous
1..1 ...1	X'91' 3277 remote
1..1 ..1.	X'92' 3275 remote
1..1 ..11	X'93' BTAM 3284 remote and VTAM 3270P
1..1 .1..	X'94' BTAM 3286 remote
1..1 1..1	X'99' 3277 local
1..1 1.11	X'9B' BTAM 3284 local
1..1 11..	X'9C' BTAM 3286 local
1.1.	X'A0' Bisynchronous - programmable
1.1. ...1	X'A1' System/3
1.1. .1..	X'A4' System/370®
1.1. ..11.	X'A6' System/7 with BSCA
1.11	X'B0' SDLC device class
1.11 ...1	X'B1' 3601

Code	Meaning
1.11 ..1.	X'B2' 3614
1.11 .1..	X'B4' 3790
1.11 .1.1	X'B5' 3790 User program
1.11 ..11.	X'B6' 3790 SCS printer
1.11 1...	X'B8' 3650 Pipeline
1.11 1..1	X'B9' 3653 Host conversational
1.11 1.1.	X'BA' 3650 Attached 3270 HC
1.11 1.11	X'BB' 3650 User program
1.11 11.1	X'BD' Contention logical unit
1.11 111.	X'BE' Interactive logical unit
1.11 1111	X'BF' Batch logical unit
11..	X'C0' LUTYPE 6

Note: An ASSIGN TERMCODE for an ISC session returns a X'C0' for LUTYPE 6. An INQUIRE CONNECTION then determines whether this ISC connection is using LUTYPE6.1 or APPC protocols.

11.. ...1	X'C1' LUTYPE 4
11.1 ...1	X'D1' ISC MM conversation
11.1 ..1.	X'D2' LUC mode group entry
11.1 ..11	X'D3' LUC session

Note: X'D3' is not used.

ASSIGN FCI

This section gives the meanings of the facility control indicator codes in the data area returned by the FCI option of the ASSIGN command.

Code	Meaning
.... ...1	X'01' TERMINAL FACILITY MASK
.... ..1.	X'02' K C P MACRO FILE MASK
.... .1..	X'04' NONTERMINAL FACILITY MASK
.... 1...	X'08' DESTINATION CONTROL TABLE
...1	X'10' AID FACILITY MASK
111.	X'E0' reserved

ASSIGN

Appendix C. Translated code for CICS commands

Application programs can be written in COBOL, C, PL/I, or assembler language, and contain CICS commands. CICS translates these programs and creates an equivalent source program where each command is now translated into a call macro or statement in the language of the original source program.

COBOL

For a COBOL application program, each command is replaced by one or more MOVE statements followed by a COBOL CALL statement.

The purpose of the MOVE statements is to assign constants to COBOL data variables; this enables constants and names to be specified as arguments to options in the commands. For example, a command such as:

```
EXEC CICS RECEIVE MAP('A') END-EXEC.
```

may be translated to:

```
MOVE '      ' TO DFHEIV0.
MOVE 'A' TO DFHC0070.
CALL 'DFHEI1' USING DFHEIV0 DFHC0070 AI.
```

Declarations for the generated variables DFHEIV0 and DFHC0070 are included automatically in the working-storage section; their names are reserved. The string within the quotation marks moved to DFHEIV0 consists of characters some of which may be unprintable. Do not use EXEC, CICS, DLI, END-EXEC, or names starting with DFH, as names for user variables.

The translator modifies the linkage section by inserting the EIB structure as the first parameter, and inserts declarations of the temporary variables that it requires into the working-storage section. It also inserts a DFHCOMMAREA if the first item in the linkage section is not a DFHCOMMAREA.

New copybook DFHEIBLC

This new copybook is a lower case version of the existing DFHEIBLK copybook.

A difference is that in DFHEIBLK the top level name is

```
01 EIBLK.
```

whereas in DFHEIBLC the top level name is

```
01 dfheiblk.
```

This is consistent with the name generated by the translator today, and also conforms to the rule that CICS reserved words should start with DFH.

C

For a C application program, each command is replaced by reassignment statements followed by a dfhexec statement that passes the parameters.

PL/I

For a PL/I application program, each command is always replaced by a DO statement, a declaration of a generated entry name, a CALL statement, and an END statement. The ENTRY declaration ensures that the appropriate conversions for argument values take place.

If a PL/I on-unit consists of a single EXEC CICS command, the command should be inside a BEGIN block, for example:

```
ON ERROR BEGIN;
EXEC CICS RETURN;
END;
```

In a similar way, if an EXEC CICS command is associated with a PL/I condition prefix, the command should be inside a BEGIN block, for example:

```
(NOZERODIVIDE): BEGIN;
EXEC CICS GETMAIN
SET(ptr-ref)
LENGTH(data-value);
END;
```

If OPTIONS(MAIN) is specified, the translator modifies the parameter list by inserting the EIB structure pointer as the first parameter. If OPTIONS(MAIN) is not specified (that is, if the program is to be link-edited to the main module), the parameter list is not modified, and it is the application programmer's responsibility to address the EIB structure in the link-edited program if access to it is required. In either case, where a program commences with a valid PL/I PROCEDURE statement, the translator inserts the declaration of the EIB structure.

Assembler language

The invocation of a CICS assembler-language application program obeys system standards, which means that on entry to the application program, registers 1, 15, 14, and 13 contain the following:

- Register 1 contains the address of the parameter list; there are at least two entries in this list, as follows:
 - Address of the EIB (EXEC interface block)
 - Address of the COMMAREA; if no COMMAREA, entry is X'00000000'
- Register 15 contains the address of the entry point
- Register 14 contains the address of the return point
- Register 13 contains the address of the save area.

All other registers are undefined.

DFHECALL macro

For an assembler-language application program, each command is replaced by an invocation of the DFHECALL macro.

This macro expands to a system-standard call sequence using registers 15, 14, 0, and 1, whose contents are:

- Register 15 contains the address of the entry point in the EXEC interface program.
- Register 14 contains the address of the return point in your application program.
- Register 0 is undefined.
- Register 1 contains the address of the parameter list.

The entry point held in register 15 is resolved in the EXEC interface processor (DFHEAI) that must be link-edited with your application program.

You can specify the exit from the application program by an EXEC CICS RETURN command in your source program. Alternatively, you can let the translator-inserted macro DFHEIRET, which has been inserted before the END statement, do it. This macro only restores the registers and returns control to the address in register 14. Note that this can be used to return from a top-level program but is not advisable from a lower-level program.

During assembly, the DFHECALL macro builds an argument list in dynamic storage, so that the application program is reentrant, and then invokes the EXEC interface program (DFHEIP). DFHEIP also obeys system standards, as described above.

In addition to the invocation of the DFHECALL macro, the translator also inserts the following macros into your source program:

DFHEIGBL

This macro sets globals if you are using EXEC DLI in either a batch or an online CICS application program. Within DFHEIGBL, if DFHEIDL is set to 1, this means that the program contains EXEC DLI commands. If DFHEIDB is set to 1, this means that the program is batch DL/I. If you are not using DL/I, it is commented and set to 0.

DFHEIENT

This macro is inserted after the first CSECT or START instruction. It performs prolog code; that is, it:

- Saves registers
- Gets an initial allocation of the storage defined by DFHEISTG)
- Sets up a base register (default register 3)
- Sets up a dynamic storage register (default register 13)
- Sets up a register to address the EIB (default register 11)

DFHEIRET

This macro performs epilog code; that is, it:

- Restores registers
DFHEIRET RCREG=*nn*, where *nn* (any register number other than 13) contains the return code to be placed in register 15 after the registers are restored.
- Returns control to the address in register 14

DFHEISTG and DFHEIEND

These macros define dynamic storage; that is, they:

- Define the storage required for the parameter list
- Define a save area.

A copybook, DFHEIBLK, containing a DSECT that describes the EIB, is also included automatically.

Note that the program must have an END statement because the translator does not otherwise insert the default macros.

The example in Figure 4 on page 385 shows a simple assembler-language application program that uses the BMS command SEND MAP to send a map to a terminal. The lower part of the figure shows the output after program INSTRUCT has been translated.

```

Source program

INSTRUCT CSECT
    EXEC CICS SEND MAP('DFH$AGA') MAPONLY ERASE
END

The above source program is translated to:

    DFHEIGBL ,                INSERTED BY TRANSLATOR
INSTRUCT CSECT
    DFHEIENT                  INSERTED BY TRANSLATOR
*    EXEC CICS SEND MAP('DFH$AGA') MAPONLY ERASE
    DFHECALL =X'1804C00008000000000046204000020',
        (CHA7,=CL7'DFH$AGA*'),(____RF,DFHEIV00)
    DFHEIRET                  INSERTED BY TRANSLATOR
    DFHEISTG                  INSERTED BY TRANSLATOR
    DFHEIEND                  INSERTED BY TRANSLATOR
END

```

Figure 4. Translated code for a CICS command

Extensions to dynamic storage

You can extend dynamic storage to provide extra storage for user variables.

You do this by defining these variables in your source program in a DSECT called DFHEISTG. The maximum amount of dynamic storage obtainable using the DFHEISTG DSECT is 65264 bytes. (Note that DFHEISTG is a reserved name.) This storage is initialized to X'00'. At translation, the translator inserts the DFHEISTG macro immediately following your DFHEISTG DSECT instruction. In this way the DSECT describes dynamic storage needed for the parameter list, for the command-level interface, and for any user variables.

The example in Figure 5 shows a simple assembler-language application program that uses such variables in dynamic storage.

```

Source program

DFHEISTG DSECT
    COPY DFH$AGA              INPUT MAP DSECT
    COPY DFH$AGB              OUTPUT MAP DSECT
MESSAGE DS CL39
INQUIRY CSECT
    EXEC CICS RECEIVE MAP('DFH$AGA')
    MVC NUMBO,KEYI
    MVC MESSAGE,=CL(L'MESSAGE)'THIS IS A MESSAGE'
    EXEC CICS SEND MAP('DFH$AGB') ERASE
END

The above source program is translated to:

    DFHEIGBL ,                INSERTED BY TRANSLATOR
DFHEISTG DSECT
    DFHEISTG                  INSERTED BY TRANSLATOR
    COPY DFH$AGA              INPUT MAP DSECT
    COPY DFH$AGB              OUTPUT MAP DSECT
MESSAGE DS CL39
INQUIRY CSECT
    DFHEIENT                  INSERTED BY TRANSLATOR
*    EXEC CICS RECEIVE MAP('DFH$AGA')
    DFHECALL =X'1802C00008000000000040900000020',
        (CHA7,=CL7'DFH$AGA*'),(____RF,DFH$AGAI)
    MVC NUMBO,KEYI
    MVC MESSAGE,=CL(L'MESSAGE)'THIS IS A MESSAGE'
*    EXEC CICS SEND MAP('DFH$AGB') ERASE
    DFHECALL =X'1804C0000800000000004E204000020',
        (CHA7,=CL7'DFH$AGB*'),(____RF,DFH$AGBO)
    DFHEIRET                  INSERTED BY TRANSLATOR
    DFHEISTG                  INSERTED BY TRANSLATOR
    DFHEIEND                  INSERTED BY TRANSLATOR
END

```

Figure 5. Translated code for user variables

Multiple base registers

The values provided by the automatic insertion of DFHEIENT may be inadequate for application programs that produce a translated output greater than 4095 bytes.

For example, the translator by default only sets up one base register (register 3), or, when the DLI translator option has been specified, the literals produced by the translator initializing the DIB could fall outside the range of that single base register.

To overcome this problem, you can prevent the translator from automatically inserting its version of the DFHEIENT macro by specifying the translator option NOPROLOG. This enables you to provide your own DFHEIENT macro with the CODEREG operand so that you can specify more than one base register. You must code your own version of the DFHEIENT macro, which can have up to three operands, in place of the first CSECT or START instruction in your source program. The three operands are as follows:

- CODEREG - base registers.
- DATAREG - dynamic-storage registers.
- EIBREG - register to address the EIB.

Translated code

For example, the source code shown in Figure 4 would become:

```
INSTRUCT DFHEIENT CODEREG=(2,3,4),  
          DATAREG=(13,5),  
          EIBREG=6  
EXEC CICS SEND  
MAP('DFH$AGA')  
MAPONLY ERASE  
END
```

The symbolic register DFHEIPLR is equated to the first DATAREG either explicitly specified or obtained by default. It is recommended that, because register 13 points to the save area defined in dynamic storage by DFHEISTG, you use register 13 as the first dynamic-storage register.

DFHEIPLR is assumed by the expansion of a CICS command to contain the value set up by DFHEIENT. You should either dedicate this register or ensure that it is restored before each CICS command.

Assembler-language programs, translated with the DLI option, have a DLI initialization call inserted after each CSECT statement. Assembler-language programs larger than 4095 bytes, that do not use the CODEREG operand of the DFHEIENT macro to establish multiple base registers, must include an LTORG statement to ensure that the literals, generated by either DFHEIENT or a DLI initialization call, fall within the range of the base register.

Note that, in general, an LTORG statement is needed for every CSECT that exceeds 4095 bytes in length.

Appendix D. Terminal control

This appendix contains general information that applies to all terminals and logical units. For more detail, see the command descriptions.

Commands and options for terminals and logical units

This section describes the commands and options that apply to terminals and logical units.

Fullword lengths

For all terminal control commands, fullword length options can be used instead of halfword length options. In particular, where the following options are used in CONVERSE, RECEIVE, or SEND, the corresponding alternative can be specified instead (except for those noted):

Option	Alternative
LENGTH	FLENGTH
TOLENGTH	TOFLENGTH
FROMLENGTH	FROMFLENGTH
MAXLENGTH	MAXFLENGTH

Application programs should be consistent in their use of fullword and halfword options on terminal control commands. The maximum value that can be specified as a parameter on any length keyword is 32767. See the *CICS Application Programming Guide* for more information.

Read from terminal or logical unit (RECEIVE)

The RECEIVE command is used to read data from a terminal or logical unit. The INTO option is used to specify the area into which the data is to be placed. Alternatively, a pointer reference can be specified in the SET option. CICS acquires an area large enough to hold the data and sets the pointer reference to the address of that data.

The contents of this area are available to the task until the next terminal I/O command. However, the area does not belong to the task and is released by CICS while processing the next request. Therefore, this area cannot be passed back to CICS for further processing.

The application can use MAXLENGTH to specify the maximum length of data that the program accepts. If the MAXLENGTH option is omitted on a RECEIVE command for which the INTO option is specified, the maximum length of data the program accepts can be specified in the LENGTH option. If the MAXLENGTH option is omitted on a RECEIVE

command for which the SET option is specified, CICS acquires enough storage to hold all the available data.

If the data exceeds the specified maximum length and the NOTRUNCATE option is specified, the remaining data is made available to satisfy subsequent RECEIVE commands. If NOTRUNCATE is not specified, the data is truncated and the LENGERR condition occurs. In this event, if the LENGTH option is specified, the named data area is set to the actual data length (before truncation occurs) when data has been received. The first RECEIVE command in a task started by a terminal does not issue a terminal control read but simply copies the input buffer, even if the data length is zero. A second RECEIVE command must be issued to cause a terminal control read.

When a PA key is defined as a print key by the system initialization parameter PRINT, and that key is pressed in response to a RECEIVE command, it has no effect on the application program. The RECEIVE command is satisfied, and the application allowed to continue, when another attention (that is, one of the other PA keys, any of the PF keys, the ENTER key, or the light pen) is made at the keyboard.

Write to terminal or logical unit (SEND)

The SEND command is used to write data to a terminal or logical unit. The options FROM and LENGTH specify respectively the data area from which the data is to be taken and the length (in bytes) of the data. For a transaction started by automatic transaction initiation (ATI), a SEND command should always precede the first RECEIVE in a transaction.

WAIT option of SEND command: Unless the WAIT option is specified also, the transmission of the data associated with the SEND command is deferred until a later event, such as a syncpoint, occurs. This deferred transmission reduces the flows of data by allowing data-flow controls to be transmitted with the data.

Transmission is not deferred for distributed transaction processing when interregion communication (IRC) is in use.

Synchronize terminal I/O for a transaction (WAIT TERMINAL)

This command is used to ensure that a terminal operation has completed before further processing occurs in a task under which more than one terminal or logical unit operation is performed. Alternatively, the WAIT option can be specified in a SEND command. (A wait is always carried out for a RECEIVE command.) Either method may cause execution of a task to be suspended. If suspension is necessary,

Terminal control

control is returned to CICS. Execution of the task is resumed when the operation is completed.

Even if the WAIT option is not specified in a SEND command, the EXEC interface program ensures that the operation is completed before issuing a subsequent RECEIVE or SEND command.

Converse with terminal or logical unit (CONVERSE)

For most terminals or logical unit types, a conversational mode of communication can be used. The CONVERSE command is used for this purpose and means that the 3650 application program communicates with the host processor. If this option is not specified, the 3650 application program cannot communicate with the host processor. In general, the CONVERSE command can be considered as a combination of a SEND command followed immediately by a WAIT TERMINAL command and then by a RECEIVE command. However, not all options of the SEND and RECEIVE commands are valid for the CONVERSE command; specific rules are given in the syntax descriptions for different devices. The TOLENGTH option is equivalent to the LENGTH option of the RECEIVE command, and the FROMLENGTH option is equivalent to the LENGTH option of the SEND command.

Send an asynchronous interrupt (ISSUE SIGNAL)

This command is used, in a transaction in receive mode, to signal to the sending transaction that a mode change is needed. The execution of the command raises the SIGNAL condition on the next SEND or RECEIVE command executed in the sending transaction, and a previously executed HANDLE CONDITION command for this condition can be used either to action the request or to ignore it.

Disconnect a switched line (ISSUE DISCONNECT)

This command is used to break a line connection between a terminal and the processor, or to break a session between VTAM logical units, when the transaction is completed. If the terminal is a buffered device, the data in the buffers is lost.

When used with a VTAM terminal, ISSUE DISCONNECT, which does not become effective until the task completes, signs off the terminal, frees the COMMAREA, clears the next TRANID, stops any BMS paging, and, if autoinstall is in effect, deletes the terminal definition.

BTAM programmable terminals

CICS Transaction Server for VSE/ESA Release 1 does not support local BTAM terminals. However, it does support transaction routing from BTAM terminals attached to a remote system.

When BTAM is used by CICS for programmable binary synchronous telecommunication line management, CICS initializes the telecommunication line with a BTAM communication line management; CICS initializes the communication read initial (TI); the terminal response must be a write initial (TI) or the equivalent. If an application program makes an input request, CICS issues a read continue (TT) to that line; if the application program makes an output request, CICS issues a read interrupt (RVI) to that line. If end of transmission (EOT) is not received on the RVI, CICS issues a read continue (TT) until the EOT is received.

The programmable terminal response to a read interrupt must be "end of transmission" (EOT). The EOT response may, however, be preceded by writes, in order to exhaust the contents of output buffers (if the input buffer size is not exceeded by this data). The input buffer size is specified by the system programmer during preparation of the TCT. CICS issues a read continue until it receives an EOT, or until the input message is greater than the input buffer (an error condition).

After receiving an EOT, CICS issues a write initial (TI) or the equivalent (depending on the type of line). The programmable terminal response must be a read initial (TI) or the equivalent.

If the application program makes another output request, CICS issues a write continue (TT) to that line. If the application program makes an input request after it has made an output request, CICS turns the line around with a write reset (TR). (CICS does not recognize a read interrupt.)

To ensure that binary synchronous terminals (for example, System/370, 1130, 2780) remain coordinated, CICS processes the data collection or data transmission transaction on any line to completion, before polling other terminals on that line.

The programmable terminal actions required for the above activity, with the corresponding user application program commands and CICS actions, are summarized in Table 19 on page 389.

Table 19. BTAM programmable terminal programming

User Application Program Command	CICS ¹ Actions	Programmable Terminal Action
RECEIVE	Read initial (TI)	Write initial (TI)
	Read continue (TT)	Write continue (TT)
SEND	Read interrupt (RVI) ²	Write reset (TR) or
	Read continue (TT) ³	Write continue Write reset
	Write initial (TI)	Read initial (TI)
SEND	Write continue (TT)	Read continue (TT)
RECEIVE	Write reset (TR) ⁴	Read continue (TT)
	Read initial (TI)	Write initial (TI)

- CICS issues the macro shown, or, if the line is switched, the equivalent. The user-written programmable terminal program must issue the equivalent of the BTAM operation shown.
- An RVI sequence is indicated by the DECFLAGS field of the data event control block (DECB) being set to X'02' and a completion code of X'7F' being returned to the event control block (ECB).
- The read continue is issued only if the EOT character is not received on the read interrupt.
- Write reset is issued only for point-to-point terminals.

Automatically initiated transactions attached to a device cause the message:

```
DFH2503 AUTO OUTPUT HAS BEEN REQ,
      PLEASE PREPARE TO RECEIVE
```

to be sent to the device, which must be prepared to receive it.

Input data is deblocked to ETX, ETB, RS, and US characters. These characters are moved with the data but are not included in the data length. Characters such as NL (new line), CR (carriage return), LF (line feed), and EM (end of message) are included as data in a CICS application program.

Teletypewriter programming

The Teletypewriter (World Trade only) uses two different control characters for print formatting, as follows:

```
< carriage return (X'22' in ITA2
   code or X'15' in EBCDIC)

= line feed (X'28' in ITA2 code
   or X'25' in EBCDIC)
```

The character < should always be used first; otherwise following characters (data) may be printed while the type bar is moving to the left.

Message format

Message begin: To start a message on a new line at the left margin, the message text must begin with X'1517' (EBCDIC). CICS recognizes the X'17' and changes it to X'25' (X'17' is an IDLE character).

Message body: To write several lines with a single transmission, the lines must be separated by X'1525', or if multiple blank lines are required, by X'152525...25'.

Message end before next input: To allow input of the next message on a new line at the left margin, the preceding message must end with X'1517'. CICS recognizes X'15' and changes the character following it to X'25'.

Message end before next output: In the case of two or more successive output messages, the "message begin" and the "message end" look the same; that is X'1517', except for the last message (see above). To make the "message end" of the preceding message distinguishable from the "message begin" of the next message, the next to last character of the "message end" must not be X'15'.

Message length

Messages for teletypewriter terminals should not exceed a length of about 3000 bytes or approximately 300 words.

Connection through VTAM

Both the TWX Model 33/35 Common Carrier Teletypewriter Exchange and the WTTY Teletypewriter (World Trade only) can be connected to CICS through BTAM, or through VTAM using NTO.

If a device is connected through VTAM using NTO, the protocols used are the same as for the 3767 logical unit, and the application program can make use of these protocols (for example, HANDLE CONDITION SIGNAL). However, the data stream is not translated to a 3767 data stream but remains as that for a TWX/WTTY.

Display device operations

In addition to the standard terminal control commands for sending and receiving data, several commands and lists are provided for use with display devices such as the 3270.

The commands are:

- Print displayed information (ISSUE PRINT)
- Copy displayed information (ISSUE COPY)
- Erase all unprotected fields (ISSUE ERASEAUP)
- Handle input without data (RECEIVE)
- Handle attention identifiers (HANDLE AID)

The lists are:

- Standard attention identifier list (DFHAID)
- Standard attribute and printer control character list (DFHBMSCA)

For devices with switchable screen sizes, the size of the screen that can be used, and the size to be used for a given transaction, are defined by CICS table generation. These values can be obtained by means of the ASSIGN command, described in "ASSIGN" on page 20.

The ERASE option should always be included in the first SEND command, to clear the screen and format it according to the transmitted data. This first SEND with ERASE also selects the screen size to be used, as specified using the RDO PROFILE option SCRNSIZE. If ERASE is omitted, the screen size is the same as its previous setting, which may be incorrect.

Use the CLEAR key outside of a transaction to set the screen to its default size.

Print displayed information (ISSUE PRINT)

ISSUE PRINT prints displayed data on the first available printer that is eligible to respond to a print request.

For a 3270 logical unit or a 3650 host-conversational (3270) logical unit, it is a printer defined by the RDO TERMINAL options PRINTER and ALTPRINTER, or by a printer supplied by the autoinstall user program.

For a 3270-display logical unit with the printer adapter feature (PRINTADAPTER(YES) option on RDO TYPETERM), used with a 3274 or 3276, it is a printer allocated by the printer authorization matrix. See *An Introduction to the IBM 3270 Information Display System*.

For a 3790 (3270-display) logical unit, it is a printer allocated by the 3790.

For a printer to be available, it must be in service and not currently attached to a task.

For a 3270 logical unit to be eligible, it must have been specified by RDO TERMINAL options PRINTER and ALTPRINTER and it must have the correct buffer capacity. If

the copy feature is also specified (COPY(YES) on the RDO TYPETERM definition) the printer must be on the same control unit.

If an ISSUE PRINT command is executed, the printer involved must be owned by the same CICS system that owns the terminal that is running the transaction.

For some 3270 displays, it is possible also to print the displayed information without using CICS. See *An Introduction to the IBM 3270 Information Display System* manual.

Copy displayed information (ISSUE COPY)

The ISSUE COPY command is used to copy the format and data contained in the buffer of a specified terminal into the buffer of the terminal that started the transaction. This command cannot be used for an LUTYPE2 connection. Both terminals must be attached to the same remote control unit. The terminal whose buffer is to be copied is identified in the TERMID option. If the terminal identifier is not valid, that is, it does not exist in the TCT, then the TERMIDERR condition occurs. The copy function to be performed is defined by the copy control character (CCC) specified in the CTLCHAR option of the ISSUE COPY command.

The WAIT option of the ISSUE COPY command ensures that the operation has been completed before control is returned to the application program.

Erase all unprotected fields (ISSUE ERASEAUP)

The ISSUE ERASEAUP command is used to erase all unprotected fields of a 3270 buffer, by the following actions:

1. All unprotected fields are cleared to nulls (X'00').
2. The modified data tags (MDTs) in each unprotected field are reset to zero.
3. The cursor is positioned to the first unprotected field.
4. The keyboard is restored.

The WAIT option of the ISSUE ERASEAUP command ensures that the operation has been completed before control is returned to the application program.

Handle input without data (RECEIVE)

The RECEIVE command with no options causes input to take place and the EIB to be updated. However, data received by CICS is not passed on to the application program and is lost. A wait is implied. Two of the fields in the EIB that are updated are described below.

Cursor position (EIBCPOSN): For every terminal control (or BMS) input operation associated with a display device, the screen cursor address (position) is placed in the EIBCPOSN field in the EIB. The cursor address is in the form of a halfword binary value and remains until updated by a new input operation.

Attention identifier (EIBAID): For every terminal control (or BMS) input operation associated with a display device, an attention identifier (AID) is placed in field EIBAID in the EIB. The AID indicates which method the terminal operator has used to initiate the transfer of information from the device to CICS; for example, the ENTER key, a program function key, the light pen, and so on. The field contents remain unaltered until updated by a new input operation.

Field EIBAID can be tested after each terminal control (or BMS) input operation to determine further processing, and a standard attention identifier list (DFHAID) is provided for this purpose. Alternatively, the HANDLE AID command can be used to pass control to specified labels when the AIDs are received.

EIBAID and EIBCPOSN are also updated at task initiation for non-ATI tasks and after each terminal control and BMS input.

Appendix E. SAA Resource Recovery

SAA Resource Recovery is the recovery element of the Systems Application Architecture (SAA) Common Programming Interface (CPI).

SAA Resource Recovery provides that architecture's alternative application programming interface (API) to EXEC CICS SYNCPOINT and EXEC CICS SYNCPOINT ROLLBACK functions in CICS. (See the *SAA Common Programming Interface-Resource Recovery Reference*, SC31-6821, for more details.) This leaves only two return codes: RR_OK and RR_BACKED_OUT.

SRRCMT

Commit call (equivalent to EXEC CICS SYNCPOINT). The return codes are:

- RR_OK
- RR_COMMITTED_OUTCOME_PENDING
- RR_COMMITTED_OUTCOME_MIXED
- RR_PROGRAM_STATE_CHECK
- RR_BACKED_OUT
- RR_BACKED_OUT_OUTCOME_PENDING
- RR_BACKED_OUT_OUTCOME MIXED

Because of the restriction, these are replaced by:

- RR_COMMITTED_OUTCOME_PENDING, RR_OK
- RR_COMMITTED_OUTCOME_MIXED, RR_OK
- RR_PROGRAM_STATE_CHECK, shown as abend code ASP2
- RR_BACKED_OUT_OUTCOME_PENDING, RR_BACKED_OUT
- RR_BACKED_OUT_OUTCOME MIXED, RR_BACKED_OUT

SRRBACK

Backout call (equivalent to EXEC CICS SYNCPOINT ROLLBACK). The return codes are:

- RR_OK
- RR_COMMITTED_OUTCOME_PENDING
- RR_COMMITTED_OUTCOME_MIXED

Because of the restriction, all these are replaced by RR_OK.

Appendix F. Common Programming Interface Communications (CPI Communications)

Common Programming Interface Communications (CPI Communications) is the communication element of the Systems Applications Architecture (SAA) Common Programming Interface (CPI).

CPI Communications in CICS provides an alternative application programming interface (API) to existing CICS Advanced Program-to-Program Communications (APPC) support. CPI Communications provides distributed transaction processing (DTP) on APPC sessions and can be used in assembler language, COBOL, PL/I, or C.

CPI Communications defines an API that can be used in APPC networks that include multiple system platforms, where the consistency of a common API is seen to be of benefit.

The CPI Communications interface can converse with applications on any system that provides an APPC API. This includes applications on CICS platforms. You may use APPC API commands on one end of a conversation and CPI Communications commands on the other. CPI Communications requires specific information (side information) to begin a conversation with a partner program. CICS implementation of side information is achieved using the partner resource which your system programmer is responsible for maintaining.

The application's calls to the CPI Communications interface are resolved by link-editing it with the CICS CPI Communications link-edit stub (DFHCPLC). You can find further guidance information in the *CICS System Definition Guide*.

CPI Communications language interfaces

The CPI Communications API is defined as a general call interface. The interface is described in the *SAA Common Programming Interface Communications Reference* manual.

Appendix G. API restrictions for distributed program link

This appendix lists the API commands, indicating whether or not they are supported in a program running in a resource region in response to a distributed program link command.

Summary of the restricted API commands

<i>Table 20. Restricted API commands</i>	
ADDRESS ALLOCATE ASSIGN CONNECT PROCESS CONVERSE EXTRACT PROCESS FREE CONVID HANDLE AID	ISSUE PURGE MESSAGE RECEIVE ROUTE SEND SIGNOFF SIGNON WAIT TERMINAL

List of API commands

The following table summarizes the CICS API commands by functional area, indicating whether or not they are supported in a program invoked by a distributed program link command. Generally, if the program issues a command that is not supported, CICS returns an INVREQ condition, with a RESP2 value of 200.

<i>Table 21 (Page 1 of 4). Summary of the CICS API by functional area</i>		
<i>Functional area</i>	<i>Command</i>	<i>Supported?</i>
Abend support	ABEND ASSIGN ABCODE ASRAINTRPT ASRAPSW ASRAREGS ORGABCODE HANDLE ABEND	YES
APPC mapped communication	ALLOCATE(APPC) CONNECT PROCESS CONVERSE EXTRACT PROCESS FREE CONVID ISSUE ABEND CONFIRMATION ERROR PREPARE SIGNAL RECEIVE SEND WAIT CONVID	NO
Note: The above APPC commands are restricted only when they refer to the principal facility.		
Signon	SIGNON SIGNOFF	NO
Batch data interchange commands	ISSUE ABORT QUERY ADD RECEIVE END REPLACE ERASE SEND NOTE WAIT	NO

API restrictions

<i>Table 21 (Page 2 of 4). Summary of the CICS API by functional area</i>		
Functional area	Command	Supported?
BMS commands	ASSIGN COLOR PAGENUM DESTCOUNT PARTNPAGE INPARTN PARTNS LDCMNEM PARTNSET LDCNUM PS MAPCOLUMN MAPHEIGHT MAPLINE MAPWIDTH OPCLASS PURGE MESSAGE RECEIVE MAP PARTN ROUTE SEND CONTROL MAP PAGE PARTNSET TEXT TEXT MAPPED TEXT NOEDIT	NO
Built-in functions	BIF DEEDIT	YES
Condition handling	HANDLE CONDITION IGNORE CONDITION PUSH HANDLE POP HANDLE	YES
Console support	WRITE OPERATOR	YES
Diagnostic services	DUMP ENTER TRACE	YES
Environmental services	ASSIGN ABCODE ABDUMP ABPROGRAM APPLID ASRAINTRPT ASRAKEY ASRAPSW ASRAREGS ASRASTG CMDSEC CWALENG INITPARM INITPARLEN INVOKINGPROG NETNAME OPERKEYS OPID ORGABCODE PRINSYSID PROGRAM RESSEC RESTART RETURNPROG STARTCODE SYSID TASKPRIORITY TCTUALENG TRANPRIORITY TWALENG USERID	YES

<i>Table 21 (Page 3 of 4). Summary of the CICS API by functional area</i>		
Functional area	Command	Supported?
Environmental services (continued)	ASSIGN ALTSCRNHT MAPLINE ALTSCRNWD MAPWIDTH APLKYBD MSRCONTROL APLTEXT NATLANGINUSE BTRANS NEXTTRANSID COLOR NUMTAB DEFSCRNHT OPCLASS DEFSCRNWD OPSECURITY DELIMITER OUTLINE DESTCOUNT PAGENUM DESTID PARTNPAGE DESTIDLENG PARTNS DSSCS PARTNSET DS3270 PS EWASUPP QNAME EXTDS SCRNHT FACILITY SCRNWD FCI SIGDATA GCHARS SOSI GCODES STATIONID GMMI TELLERID HIGHLIGHT TERMCODE INPARTN TERMPRIORITY KATAKANA TEXTKYBD LANGINUSE TEXTPRINT LDCMNEM UNATTEND LDCNUM USERNAME MAPCOLUMN USERPRIORITY MAPHEIGHT VALIDATION	NO
Environmental services (continued)	ADDRESS ACEE TWA COMMAREA CWA EIB TCTUA	YES
File control	DELETE ENDBR READ READNEXT READPREV RESETBR REWRITE STARTBR UNLOCK WRITE	YES
Interval control	ASKTIME CANCEL DELAY FORMATTIME POST RETRIEVE START WAIT EVENT	YES
Journaling	WRITE JOURNALNUM WAIT JOURNALNUM	YES
Monitoring	MONITOR POINT	YES
Program control	LINK LOAD RELEASE RETURN XCTL	YES
Note: LINK, RETURN and XCTL do not support INPUTMSG.		

API restrictions

<i>Table 21 (Page 4 of 4). Summary of the CICS API by functional area</i>		
Functional area	Command	Supported?
Security	QUERY SECURITY	YES
Storage control	FREEMAIN GETMAIN	YES
Syncpoint	SYNCPPOINT	YES
Task control	ASSIGN TASKPRIORITY CHANGE TASK DEQ ENQ SUSPEND WAIT EXTERNAL	YES
Temporary storage	DELETEQ TS READQ TS WRITEQ TS	YES
Terminal control	ASSIGN FACILITY CONVERSE HANDLE AID RECEIVE SEND WAIT TERMINAL	NO
Transient data	DELETEQ TD READQ TD WRITEQ TD	YES

Appendix H. CVDA numeric values

This appendix lists the CVDA values and their numeric equivalents for the relevant EXEC CICS commands in this book.

Table 22. CVDAs and numeric values in alphabetic sequence

cvda	value
ALLOCATED	81
ALTERABLE	52
CICS	660
CICSEXECKEY	381
CONFFREE	82
CONFRECEIVE	83
CONFSEND	84
CRITICAL	11
CTRLABLE	56
EVENTUAL	3
FREE	85
IMMEDIATE	2
LOG	54
LUW	246
NOLOG	55
NONCICS	661
NOTALTERABLE	53
NOTAPPLIC	1
NOTCTRLABLE	57
NOTPURGEABLE	161
NOTREADABLE	36
NOTUPDATABLE	38
PENDFREE	86
PENDRECEIVE	87
PURGEABLE	160
READABLE	35
READONLY	275
RECEIVE	88
ROLLBACK	89
SEND	90
SYNCFREE	91
SYNCRECEIVE	92
SYNCSEND	93
TASK	233
UPDATABLE	37
USER	642
USEREXECKEY	382

Table 23. CVDAs and numeric values in numeric sequence

value	cvda
1	NOTAPPLIC
2	IMMEDIATE
3	EVENTUAL
11	CRITICAL
35	READABLE
36	NOTREADABLE
37	UPDATABLE
38	NOTUPDATABLE
52	ALTERABLE
53	NOTALTERABLE
54	LOG
55	NOLOG
56	CTRLABLE
57	NOTCTRLABLE
81	ALLOCATED
82	CONFFREE
83	CONFRECEIVE
84	CONFSEND
85	FREE
86	PENDFREE
87	PENDRECEIVE
88	RECEIVE
89	ROLLBACK
90	SEND
91	SYNCFREE
92	SYNCRECEIVE
93	SYNCSEND
160	PURGEABLE
161	NOTPURGEABLE
233	TASK
246	LUW
275	READONLY
381	CICSEXECKEY
382	USEREXECKEY
642	USER
660	CICS
661	NONCICS

CVDA numeric values

Appendix I. National language codes

Language codes are held as one character for NATLANG and NATLANGINUSE, and three characters for LANGUAGECODE and LANGINUSE.

Table 24. CICS language suffixes

Suffix	IBM Code	Language name
A	ENG	UK English
B	PTB	Brazilian Portuguese
C	CHS	Simplified Chinese
D	DAN	Danish
E	ENU	US English
F	FRA	French
G	DEU	German
H	KOR	Korean
I	ITA	Italian
J	ISL	Icelandic
K	JPN	Japanese
L	BGR	Bulgarian
M	MKD	Macedonian
N	NOR	Norwegian
O	ELL	Greek
P	PTG	Portuguese
Q	ARA	Arabic
R	RUS	Russian
S	ESP	Spanish
T	CHT	Traditional Chinese
U	UKR	Ukrainian
V	SVE	Swedish
W	FIN	Finnish
X	HEB	Hebrew
Y	SHC	Serbo-Croatian (Cyrillic)
Z	THA	Thai
1	BEL	Byelorussian
2	CSY	Czech
3	HRV	Croatian
4	HUN	Hungarian
5	PLK	Polish
6	ROM	Romanian
7	SHL	Serbo-Croatian (Latin)
8	TRK	Turkish
9	NLD	Dutch

There are other IBM codes not supported by CICS.

Table 25. Other IBM language codes

IBM Code	Language name
AFR	Afrikaans
CAT	Catalan
DES	Swiss German
ENA	Australian English
ENP	English Upper Case
FRB	Belgian French
FRC	Canadian French
FRS	Swiss French
GAE	Irish Gaelic
ITS	Swiss Italian
NLB	Belgian Dutch - Flemish
NON	Norwegian - Nynorsk
RMS	Rhaeto-Romanic
SKY	Slovakian
SLO	Slovenian
SRL	Serbian (Latin)
SRB	Serbian (Cyrillic)
SQI	Albanian
URD	Urdu

National language codes

Appendix J. BMS-related constants

This appendix contains the BMS-related standard attribute and printer control characters, a bit map for attributes, MSR control value constants, and attention identifier constants.

The standard list DFHBMSCA makes it simpler to provide field attributes and printer control characters. Table 26 lists the symbolic names for the various combinations of attributes and control characters. If you need combinations other than the ones shown, you must generate them separately. To help you do this, see Table 27 on page 407 for a bit map of attributes. To find the value of an attribute constant, see the *3274 Control Unit Reference Summary*.

You can get the standard attribute and printer character control list by copying copybook DFHBMSCA into your application.

- For COBOL users, it consists of a set of 01 statements that can be copied into the working storage section.
- For C users, it is included in applications as follows:

```
#include "dfhbmsca.h"
```

- For PL/I users, it consists of DECLARE statements defining elementary character variables.
- For assembler-language users, the list consists of a set of EQU statements.

You must use the symbolic name DFHDFT in the application structure to override a map attribute with the default. You can use a high value, such as X'FF', to reset the COLOR, HIGHLIGHT, OUTLINE, PS, SOSI, or VALIDN attributes to their default values. On the other hand, to specify default values in a set attribute (SA) sequence in text build, you should use the symbolic names DFHDFCOL, DFHBASE, or DFHDFHI.

Table 26 (Page 1 of 2). Standard attribute and printer control character list, DFHBMSCA

Constant	Meaning
DFHBMPER	Printer end-of-message
DFHBMPNL	Printer new-line
DFHBMPFF	Printer form feed
DFHBMPCR	Printer carriage return
DFHBMASK	Autoskip
DFHBMUNP	Unprotected
DFHBMUNN	Unprotected and numeric
DFHBMPRO	Protected
DFHBMBRY	Bright
DFHBMNDAR	Dark
DFHBMFSE	MDT set
DFHBMPRF	Protected and MDT set
DFHBMASF	Autoskip and MDT set
DFHBMASB	Autoskip and bright
DFHBMPSO	shift-out value X'0E'.
DFHBMPSI	shift-in value X'0F'.
DFHBMEOF	Field erased
DFHBMCUR	Field containing cursor flagged
DFHBMEC	Erased field containing cursor (COBOL only)
DFHBMFLG	Flags (COBOL only)
DFHBMDET	Field detected
DFHSA ¹	Set attribute (SA) order
DFHERROR	Error code
DFHCOLOR ¹	Color
DFHPS ¹	Programmed symbols
DFHHLT ¹	Highlight
DFH3270 ¹	Base 3270 field attribute
DFHVAL	Validation
DFHOUTLN	Field outlining attribute code
DFHBKTRN	Background transparency attribute code
DFHALL ¹	Reset all to defaults
DFHDFT	Default
DFHDFCOL ¹	Default color
DFHBLUE	Blue
DFHRED	Red
DFHPINK	Pink
DFHGREEN	Green
DFHTURQ	Turquoise
DFHYELLO	Yellow

Table 26 (Page 2 of 2). Standard attribute and printer control character list, DFHBMSCA

Constant	Meaning
DFHNEUTR	Neutral
DFHBASE ¹	Base programmed symbols
DFHDFHI ¹	Normal
DFHBLINK	Blink
DFHREVRS	Reverse video
DFHUNDLN	Underscore
DFHMFIL ²	Mandatory fill
DFHMENT ²	Mandatory enter
DFHMFE	Mandatory fill and mandatory enter
DFHMT	Trigger
DFHMFT	Mandatory fill and trigger
DFHMET	Mandatory enter and trigger
DFHMFET	Mandatory fill and mandatory enter and trigger
DFHUNNOD	Unprotected, nondisplay, nonprint, nondetectable, MDT
DFHUNIMD	Unprotected, intensify, light-pen detectable, MDT
DFHUNNUM	Unprotected, numeric, MDT
DFHUNNUB	Unprotected, numeric, intensify, intensify, light-pen detectable
DFHUNINT	Unprotected, numeric, intensify, light-pen detectable, MDT
DFHUNNON	Unprotected, numeric, nondisplay, nonprint, nondetectable, MDT
DFHPROTI	Protected, intensify, light-pen detectable
DFHPROTN	Protected, nondisplay, nonprint, nondetectable
DFHDFFR	Default outline
DFHUNDER	Underline
DFHRIGHT	Right vertical line
DFHOVER	Overline
DFHLEFT	Left vertical line
DFHBOX	Underline and right vertical and overline and left vertical
DFHSOSI	SOSI=yes
DFHTRANS	Background transparency
DFHOPAQ	No background transparency
Notes:	
¹ For text processing only. Use for constructing embedded set attribute orders in user text.	
² Cannot be used in set attribute orders.	

Table 27. Bit map for attributes

prot	a/n	hi	spd	ndp	mdt	ebcd	ascii	char
U						40	20	b (blank)
U					Y	C1	41	A
U			Y			C4	44	D
U			Y		Y	C5	45	E
U		H	Y			C8	48	H
U		H	Y		Y	C9	49	I
U				Y		4C	3C	<
U				Y	Y	4D	28	(
U	N					50	26	
U	N				Y	D1	4A	J
U	N		Y			D4	4D	M
U	N		Y		Y	D5	4E	N
U	N	H	Y			D8	51	Q
U	N	H	Y		Y	D9	52	R
U	N			Y		5C	2A	*
U	N			Y	Y	5D	29)
P						60	2D	- (hyphen)
P					Y	61	2F	/
P			Y			E4	55	U
P			Y		Y	E5	56	V
P		H	Y			E8	59	Y
P		H	Y		Y	E9	5A	Z
P				Y		6C	25	%
P				Y	Y	6D	5F	_ (underscore)
P	S					F0	30	0
P	S				Y	F1	31	1
P	S		Y			F4	34	4
P	S		Y		Y	F5	35	5
P	S	H	Y			F8	38	8
P	S	H	Y		Y	F9	39	9
P	S			Y		7C	40	@
P	S			Y	Y	7D	27	,

The attributes in the headings are:

prot = protected
 a/n = automatic skip or numeric
 hi = high intensity
 spd = selector pen detectable
 ndp = nondisplay print
 mdt = modified data tag

The hexadecimal codes in the headings are:

ebcd = extended binary-coded decimal interchange code
 ascii = American National Standard Code for Information Interchange
 char = graphic character equivalent to hex code

The characters in the body of the above table mean the following:

H = High
 P = Protected
 N = Numeric
 U = Unprotected
 S = Automatic skip
 Y = Yes

Magnetic slot reader (MSR) control value constants, DFHMSRCA

A selection of MSR control value constants has been created for CICS and stored in copybook DFHMSRCA. The patterns are stored as named constants that can be loaded by simple application program commands. Provision of such constants saves the programmer from having to build a commonly used bit pattern whenever it is required.

MSR control byte values

A selection of MSR control byte values has been created for CICS and stored in the copybook DFHMSRCA. Table 28 on page 408 shows the meaning of each bit. The constants supplied in DFHMSRCA are listed in Table 29 on page 408.

Table 28. MSR control byte values

Byte (purpose)	Bit	Value	Meaning
1 (STATE MASK) If a bit is on in the STATE MASK byte, the state it represents is adopted by the device if the corresponding bit is also on in the STATE VALUE byte.	0	USER	User mode. Turn the yellow light on if the same bit is on in STATE VALUE.
	1	LOCK	Locked/Unlocked. If locked, MSR input is inhibited.
	2	AUTO	Autoenter on/off. If set on, any card read by the MSR causes an ENTER operation. If off, only a secure card causes an ENTER.
	3	Ai1S	Suppress audible alarm 1.
	4	Ai2S	Suppress audible alarm 2.
2 (STATE VALUE) Modifies state to on or off if the corresponding bit is set on in STATE MASK.			
3 (INDICATOR MASK) Performs a similar function to STATE MASK, but for indicators.	0		Light 1 (Green)
	1		Light 2 (Yellow)
	2		Light 3 (Red)
	3		Audible Alarm 1 (Long buzz)
	4		Audible Alarm 2 (Short buzz)
4 (INDICATOR VALUE) Performs a similar function to STATE VALUE.			

Table 29. Standard list DFHMSRCA

Constant	Meaning
DFHMSRST	MSR reset. All lights and buzzers off. MSR available for input.
DFHMSSCON	Transaction ready for more input. Green and yellow on; emit short buzz; IN PROCESS (user) mode set.
DFHMSSFIN	Input complete. Green on; emit short buzz; IN PROCESS mode reset.
DFHMSSALR	Operator alert. Green, yellow, and red on; emit long buzz; IN PROCESS mode reset.
DFHMSSALS	Operator alert. Green, yellow, and red on; emit long buzz; IN PROCESS mode set.
DFHMSSIPY	IN PROCESS state set. Yellow on.
DFHMSSIPN	IN PROCESS state reset.
DFHMSSLKY	MSR operation inhibited. Yellow on.
DFHMSSLKN	MSR input allowed. Green on. Yellow on.
DFHMSSAEY	MSR autoenter on. Yellow on.
DFHMSSAEN	MSR autoenter off. Yellow on.
DFHMSSLBN	Long buzzer suppressed. Yellow on.
DFHMSSLBY	Long buzzer permitted. Yellow on.
DFHMSSBN	Short buzzer suppressed. Yellow on.
DFHMSSBY	Short buzzer permitted. Yellow on.
DFHMSSNOP	Leave all MSR settings unchanged.

Attention identifier constants, DFHAID

The standard attention identifier list, DFHAID, simplifies testing the contents of the EIBAID field. Table 30 shows the symbolic name for the attention identifier (AID) and the corresponding 3270 function.

You can get a copy of the list by copying DFHAID into your application program. For COBOL users, it consists of a set of 01 statements that must be copied into the working-storage section. For C users, it consists of a series of defined constants. For PL/I users, it consists of DECLARE statements defining elementary character variables.

<i>Table 30. Standard list DFHAID</i>	
Constant	Meaning
DFHENTER	ENTER key.
DFHCLEAR	CLEAR key.
DFHPA1–DFHPA3	PA1–PA3 keys.
DFHPPF1–DFHPPF24	PF1–PF24 keys.
DFHOPIID	OPERID or MSR.
DFHMSRE	Extended (standard) MSR.
DFHTRIG	Trigger field.
DFHPEN	SELECTOR PEN or CURSOR SELECT key.
DFHCLRP	CLEAR PARTITION key.
DFHSTRF	Structured field pseudo-AID.
Note: DFHCLRP and DFHSTRF do not apply to minimum function BMS.	

Appendix K. BMS macro summary

This appendix contains the syntax of each BMS macro, separating the various operands and options into those appropriate to minimum, standard, and full BMS.

When coding, you should have the title in column 1, the macro in column 10, continuation lines should have * in column 72 and continue on column 16 on the next line.

For more information about BMS, see the *CICS Application Programming Guide*.

Mapset, map, and field definition

You should ensure that the names of maps, and names of fields within a mapset (or within multiple mapsets that are copied into one application program) are unique. Maps and mapsets must not have the same name.

You assemble a BMS mapset definition to generate either a symbolic description map or a physical map. The physical map is a structured data area used at execution time to build the data stream for the terminal. The symbolic map is a series of data structures which you copy into your program at compile time so you can refer to the fields in the map by name.

Before CICS can load a physical map, you must define it to CICS using DEFINE TRANSACTION with the MAPSET attribute.

An alternative to defining maps using RDO is to use the program autoinstall exit to create the definition when the mapset is first used. (For programming information about the autoinstall user program, see the *CICS Customization Guide*.) The *CICS System Definition Guide* tells you how to assemble and catalog the maps.

Mapset definition macro (DFHMSD)

The DFHMSD macro defines a mapset.

Map definition macro (DFHMDI)

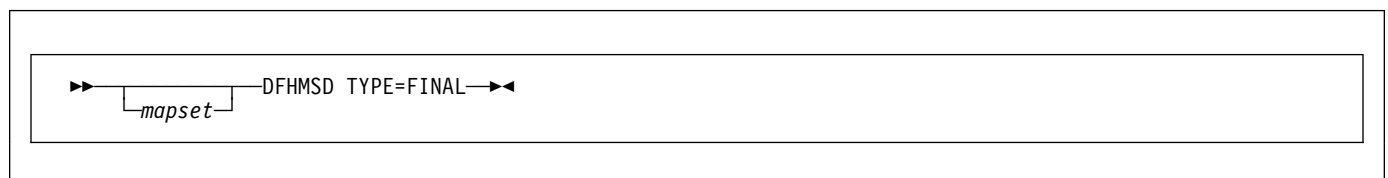
The DFHMDI macro defines a map within the mapset defined by the previous DFHMSD macro. A map contains zero or more fields.

Field definition macro (DFHMDF)

The DFHMDF macro defines a field within a map defined by the previous DFHMDI macro.

Ending a mapset definition

A mapset definition ends with a macro of the form:



“mapset” is optional, but if used it must be the same as that on the DFHMSD macro that began the mapset definition.

Partition set definition

Partitions are defined by coding the macros DFHPSD (partition set definition) and DFHPDI (partition definition). Each partition definition must be part of a partition set definition.

BMS macro summary

Partition set definition macro (DFHPSD)

Each partition set definition contains a single DFHPSD macro followed by one or more DFHPDI macros, and ending with a partition set definition TYPE=FINAL.

Before CICS can load a physical map, you must define the physical map to CICS with an RDO MAPSET resource definition.

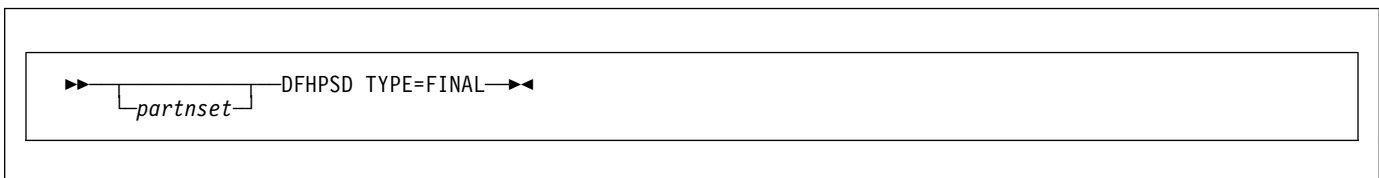
An alternative to defining maps using RDO is to use the PROGRAM autoinstall exit to create the definition when the mapset is first used. (For programming information about the PROGRAMautoinstall user program, *CICS Customization Guide*.)

Partition definition macro (DFHPDI)

A partition set contains one or more partitions. Each partition is defined by coding a partition definition macro.

Ending a partition set definition

A partition set definition ends with a macro of the form:



The PARTNSET name (if specified) must match that specified on the DFHPSD macro that started the partition set definition.

Field groups

Very often, an output data display field has to contain several subfields, all sharing the same display attributes, and each of which might have to be modified separately. At output, subfields that have not been modified by the program can adopt default data values from the output map. For example, a display can include a date field of a “day” subfield, “month” subfield, and “year” subfield. The contents of the year subfield remain constant over a relatively long period; its value can safely be taken from a map. However, the day value and month value must be updated regularly. Similarly, on input the terminal operator can enter data in each subfield separately.

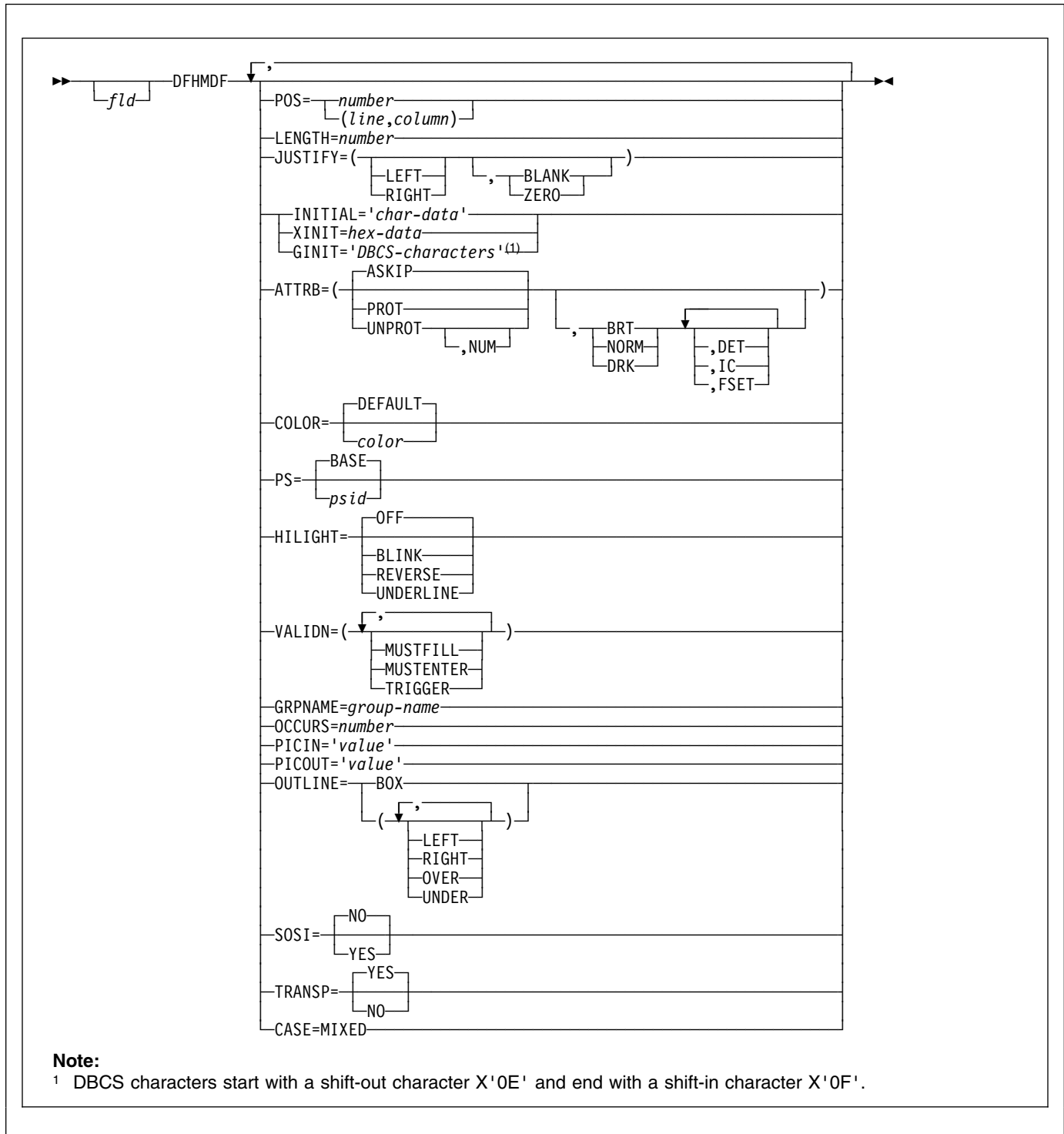
You use the GRPNAME operand to define a group of subfields that combine to produce a field. The start of the group is indicated by a DFHMDF macro with the GRPNAME operand. This operand defines the first subfield, and specifies the attributes and name of the group. It is followed by other DFHMDF macros, one for each of the other subfields. Each of these must specify the group name, but cannot specify attribute values. The definition of the group is terminated by a DFHMDF macro that specifies a different group name, by one that specifies no group name, or by a DFHMDI or DFHMSD macro.

Briefly, a group of fields in a map would appear as follows in the map definition:

```
MAPSET DFHMSD....
      .
      .
MAP    DFHMDI....
      .
      .
DD     DFHMDF GRPNAME=DATE,POS=40,
        LENGTH=2,ATTRB=...
      .
MM     DFHMDF GRPNAME=DATE,POS=46,
        LENGTH=2
      .
YY     DFHMDF GRPNAME=DATE,POS=52,
        LENGTH=2
      .
FIELD  DFHMDF LENGTH=5,COLOR=GREEN,...
        DFHMSD TYPE=FINAL
```

The POS operand specifies the position of the attribute byte of the field even though subfields of a group, other than the first, do not have attributes. If the subfields are positioned contiguously with no intervening blanks, the POS of the second and succeeding subfields must specify the position of the last character of the previous subfield.

DFHMDF



Description

The DFHMDF macro defines a field within a map defined by the previous DFHMDF macro. A map contains zero or more fields.

"fld" is the name (1–30 characters) of the field. You should, however, refer to your compiler manual to make sure that there are no other restrictions on the length.

For more information about defining field names, see the *CICS Application Programming Guide*. If "fld" is omitted, application programs cannot access the field to change its attributes or alter its contents. For an output map, omitting the field name may be appropriate when the INITIAL operand is used to specify the contents of a field. If a field name is specified and the map that includes the field is used in a mapping operation, nonnull data supplied by the user overlays data supplied by initialization (unless default data only is being written).

The performance of input mapping operations is optimized if DFHMDF macros are arranged in numerical order of the POS operand.

You cannot define more than 1023 named fields for a COBOL, C, or PL/I input/output map.

You should ensure that the names of maps, and names of fields within a mapset (or within multiple mapsets that are copied into one application program) are unique. Maps and mapsets must not have the same name.

Before CICS can load a physical map, you must define the physical map to CICS with an RDO MAPSET resource definition.

operands

ATTRB

This operand applies only to 3270 data stream devices; it is ignored for other devices, except that ATTRB=DRK is honored for the SCS Printer Logical Unit. It is also ignored (except for ATTRB=DRK) if the NLEOM option is specified on the SEND MAP command for transmission to a 3270 printer. In particular, ATTRB=DRK should not be used as a method of protecting secure data on output on non-3270, non-SCS printer terminals. Refer to *An Introduction to the IBM 3270 Information Display System* for further information.

If ATTRB is specified within a group of fields, it must be specified in the first field entry. A group of fields appears as one field to the 3270. Therefore, the ATTRB specification refers to all of the fields in a group as one field rather than as individual fields. It specifies device-dependent characteristics and attributes, such as the capability of a field to receive data, or the intensity to be used when the field is output. It could however, be used for making an input field nondisplay for secure entry of a password from a screen. For input map fields,

DET and NUM are the only valid options; all others are ignored.

ASKIP is the default and specifies that data cannot be keyed into the field and causes the cursor to skip over the field.

BRT specifies that a high-intensity display of the field is required. Because of the 3270 attribute character bit assignments, a field specified as BRT is also potentially detectable. However, for the field to be recognized as detectable by BMS, DET must also be specified.

DET specifies that the field is potentially detectable.

The first character of a 3270 detectable field must be one of the following:

? > & blank

If ? or >, the field is a selection field; if & or blank, the field is an attention field. (See *An Introduction to the IBM 3270 Information Display System* for further details about detectable fields.)

A field for which BRT is specified is potentially detectable to the 3270, because of the 3270 attribute character bit assignments, but is not recognized as such by BMS unless DET is also specified.

DET and DRK are mutually exclusive.

If DET is specified for a field on a map with MODE=IN, only one data byte is reserved for each input field. This byte is set to X'00', and remains unchanged if the field is not selected. If the field is selected, the byte is set to X'FF'.

No other data is supplied, even if the field is a selection field and the ENTER key has been pressed.

If the data in a detectable field is required, all of the following conditions must be fulfilled:

1. The field must begin with one of the following characters:
? > & blank
and DET must be specified in the output map.
2. The ENTER key (or some other attention key) must be pressed after the field has been selected, although the ENTER key is not required for detectable fields beginning with & or a blank.
3. DET must not be specified for the field in the input map. DET must, however,

be specified in the output map. For more information about BMS support of the light pen, see the *CICS Application Programming Guide*.

- DRK** specifies that the field is nonprint/nondisplay. DRK cannot be specified if DET is specified.
- FSET** specifies that the modified data tag (MDT) for this field should be set when the field is sent to a terminal.
- Specification of FSET causes the 3270 to treat the field as though it has been modified. On a subsequent read from the terminal, this field is read, whether or not it has been modified. The MDT remains set until the field is rewritten without ATTRB=FSET, or until an output mapping request causes the MDT to be reset.
- Either of two sets of defaults may apply when a field to be displayed on a 3270 is being defined but not all parameters are specified. If no ATTRB parameters are specified, ASKIP and NORM are assumed. If any parameter is specified, UNPROT and NORM are assumed for that field unless overridden by a specified parameter.
- IC** specifies that the cursor is to be placed in the first position of the field. The IC attribute for the last field for which it is specified in a map is the one that takes effect. If not specified for any fields in a map, the default location is zero. Specifying IC with ASKIP or PROT causes the cursor to be placed in an unkeyable field.
- This option can be overridden by the CURSOR option of the SEND MAP command that causes the write operation.
- NORM** specifies that the field intensity is to be normal.
- NUM** ensures that the data entry keyboard is set to numeric shift for this field unless the operator presses the alpha shift key, and prevents entry of nonnumeric data if the Keyboard Numeric Lock feature is installed.
- PROT** specifies that data cannot be keyed into the field.
- If data is to be copied from one device to another attached to the same 3270 control unit, the first position (address 0) in the buffer of the device to be copied from must not contain an attribute byte for a protected field. Therefore, when preparing maps for 3270s, ensure that the first map of any page does not contain a protected field starting at position 0.

UNPROT specifies that data can be keyed into the field.

CASE

specifies that the field contains both uppercase and lowercase data that is to be converted to uppercase if the terminal definition specifies katakana support (KATAKANA(YES) option on RDO TYPETERM definition).

This should be specified if a field is known to contain lowercase Latin characters but may be displayed on a katakana display. It should not be specified if the field may contain valid katakana characters.

COLOR

indicates the individual color, or the default color for the mapset (where applicable).

The valid colors are blue, red, pink, green, turquoise, yellow, and neutral.

The COLOR operand is ignored unless the terminal supports color, as indicated by the RDO TYPETERM option COLOR(YES).

GINIT

specifies constant or default data for an output field. GINIT is used to specify data in DBCS character strings, which must be enclosed by SO (shift out, X'0E') and SI (shift in, X'0F') characters. When GINIT is specified, the length must be even and is the number of bytes in the string (that is, not the number of DBCS characters). If a graphic data type (PS=X'F8') is used, and the language is COBOL2, a PIC G is generated. Only one of GINIT, INITIAL, or XINIT may be specified.

GRPNAME

is the name used to generate symbolic storage definitions and to combine specific fields under one group name. The same group name must be specified for each field that is to belong to the group. The length of the name is up to 30 characters though you should refer to your compiler manual to make sure that there are no other restrictions on the length.

If this operand is specified, the OCCURS operand cannot be specified.

The fields in a group must follow on; there can be gaps between them, but not other fields from outside the group. A field name must be specified for every field that belongs to the group, and the POS operand must also be specified to ensure that the fields follow each other. All the DFHMDF macros defining the fields of a group must be placed together, and in the correct order (ascending numeric order of the POS value).

For example, the first 20 columns of the first six lines of a map can be defined as a group of six fields, as long as the remaining columns on the first five lines are not defined as fields.

The ATTRB operand specified on the first field of the group applies to all of the fields within the group.

HIGHLIGHT

specifies the default highlighting attribute for all fields in all maps in a mapset.

OFF is the default and indicates that no highlighting is used.

BLINK specifies that the field must blink.

REVERSE specifies that the character or field is displayed in reverse video, for example, on a 3278, black characters on a green background.

UNDERLINE specifies that a field is underlined.

The HIGHLIGHT operand is ignored unless the terminal supports highlighting, as indicated by the RDO TYPETERM option HIGHLIGHT(YES).

INITIAL (or XINIT)

specifies constant or default data for an output field. INITIAL is used to specify data in character form; XINIT is used to specify data in hexadecimal form.

For fields with the DET attribute, initial data that begins with one of the following characters:

? > & blank

should be supplied.

The number of characters that can be specified in the INITIAL operand is restricted to the continuation limitation of the assembler to be used or to the value specified in the LENGTH operand (whichever is the smaller).

Hexadecimal data is written as an even number of hexadecimal digits, for example, XINIT=C1C2. If the number of valid characters is smaller than the field length, the data is padded on the right with blanks. For example, if LENGTH=3, XINIT=C1C2 results in an initial field of 'AB'.

If hexadecimal data is specified that corresponds with line or format control characters, the results are unpredictable. The XINIT operand should therefore be used with care. Only one of GINIT, INITIAL, or XINIT may be specified.

JUSTIFY

specifies the field justifications for input operations. This operand is ignored for VTAM-supported 3600, 3650, and 3790 terminals, because input mapping is not available.

LEFT specifies that data in the input field is left-adjusted.

RIGHT specifies that data in the input field is right-adjusted.

BLANK specifies that blanks are to be inserted in any unfilled positions in an input field.

ZERO specifies that zeros are to be inserted in any unfilled positions in an input field.

LEFT and RIGHT are mutually exclusive, as are BLANK and ZERO. If certain arguments are supplied but others are not, assumptions are made as follows:

Specified	Assumed
LEFT	BLANK
RIGHT	ZERO
BLANK	LEFT
ZERO	RIGHT

If JUSTIFY is omitted, but the NUM attribute is specified, RIGHT and ZERO are assumed. If JUSTIFY is omitted, but attributes other than NUM are specified, LEFT and BLANK are assumed.

Note: If a field is initialized by an output map or contains data from any other source, data that is typed as input overwrites only the equivalent length of the existing data; surplus existing data remains in the field and could cause unexpected interpretation of the new data.

LENGTH

specifies the length (1–256 bytes) of the field or group of fields. This length should be the maximum length required for application program data to be entered into the field; it should not include the one-byte attribute indicator appended to the field by CICS for use in subsequent processing. The length of each individual subfield within a group must not exceed 256 bytes. LENGTH can be omitted if PICIN or PICOUT is specified, but is required otherwise. You can specify a length of zero only if you omit the label (field name) from the DFHMDF macro. That is, the field is not part of the application data structure and the application program cannot modify the attributes of the field. You can use a field with zero length to delimit an input field on a map.

The map dimensions specified in the SIZE operand of the DFHMDF macro defining a map can be smaller than the actual page size or screen size defined for the terminal.

If the LENGTH specification in a DFHMDF macro causes the map-defined boundary on the same line to be exceeded, the field on the output screen is continued by wrapping.

OCCURS

specifies that the indicated number of entries for the field are to be generated in a map, and that the map definition is to be generated in such a way that the fields are addressable as entries in a matrix or an array. This permits several data fields to be addressed by the same name (subscripted) without generating a unique name for each field.

OCCURS and GRPNAME are mutually exclusive; that is, OCCURS cannot be used when fields have been defined under a group name. If this operand is omitted, a value of OCCURS=1 is assumed.

DFHMDF

OUTLINE

allows lines to be included above, below, to the left, or to the right of a field. You can use these lines in any combination to construct boxes around fields or groups of fields.

PICIN (COBOL and PL/I only)

specifies a picture to be applied to an input field in an IN or INOUT map; this picture serves as an editing specification that is passed to the application program, thus permitting the user to exploit the editing capabilities of COBOL or PL/I. BMS checks that the specified characters are valid picture specifications for the language of the map.

However, the validity of the input data is not checked by BMS or the high-level language when the map is used, so any desired checking must be performed by the application program. The length of the data associated with "value" should be the same as that specified in the LENGTH operand if LENGTH is specified. If both PICIN and PICOUT are used, an error message is produced if their calculated lengths do not agree; the shorter of the two lengths is used. If PICIN or PICOUT is not coded for the field definition, a character definition of the field is automatically generated regardless of other operands that are coded, such as ATTRB=NUM.

As an example, assume the following map definition is created for reference by a COBOL application program:

```
MAPX DFHMSD TYPE=DSECT,
      LANG=COBOL,
      MODE=INOUT
MAP DFHMDI LINE=1,COLUMN=1,
      SIZE=(1,80)
F1 DFHMDF POS=0,LENGTH=30
F2 DFHMDF POS=40,LENGTH=10,
      PICOUT='$$$,$$0.00'
F3 DFHMDF POS=60,LENGTH=6,
      PICIN='9999V99',
      PICOUT='ZZ9.99'
      DFHMSD TYPE=FINAL
```

This generates the following DSECT:

```
01 MAPI.
  02 F1L PIC S9(4) COMP.
  02 F1A PIC X.
  02 FILLER REDEFINES F1A.
    03 F1F PIC X.
  02 F1I PIC X(30).
  02 FILLER PIC X.
  02 F2L PIC S9(4) COMP.
  02 F2A PIC X.
  02 FILLER REDEFINES F2A.
    03 F2F PIC X.
  02 F2I PIC X(10).
  02 FILLER PIC X.
  02 F3L PIC S9(4) COMP.
  02 F3A PIC X.
  02 FILLER REDEFINES F3A.
    03 F3F PIC X.
  02 F3I PIC 9999V99.
  02 FILLER PIC X.
```

```
01 MAPO REDEFINES MAPI.
  02 FILLER PIC X(3).
  02 F10 PIC X(30).
  02 FILLER PIC X.
  02 FILLER PIC X(3).
  02 F20 PIC $$$,$$0.00.
  02 FILLER PIC X.
  02 FILLER PIC X(3).
  02 F30 PIC ZZ9.99.
  02 FILLER PIC X.
```

Note: The valid picture values for COBOL input maps are:

A P S V X 9 / and (

The valid picture values for PL/I input maps are:

A B E F G H I K M P R S T V
X Y and Z

1 2 3 6 7 8 9 / + - , . *
\$ and (

Refer to the appropriate language reference manual for the correct syntax of the PICTURE attribute.

PICOUT (COBOL and PL/I only)

is similar to PICIN, except that a picture to be applied to an output field in the OUT or INOUT map is generated.

The valid picture values for COBOL output maps are:

A B E P S V X Z 0 9 , . + - \$
CR DB / and (

The valid picture values for PL/I output maps are:

A B E F G H I K M P R S T V
X Y and Z

1 2 3 6 7 8 9 / + - , . * \$
CR DB and (

Refer to the appropriate language reference manual for the correct syntax of the PICTURE attribute.

Note: COBOL supports multiple currency signs and multi-character currency signs in PICTURE specifications.

The default currency picture symbol is the dollar sign (\$), which represents the national currency symbol; for example the dollar (\$), the pound (£), or the yen (¥).

The default currency picture symbol may be replaced by a different currency picture symbol that is defined in the SPECIAL NAMES clause. The currency sign represented by the picture symbol is defined in the same clause. For example:

SPECIAL NAMES.
 CURRENCY SIGN IS '\$' WITH PICTURE SYMBOL '\$'.
 CURRENCY SIGN IS '£' WITH PICTURE SYMBOL '£'.
 CURRENCY SIGN IS 'EUR' WITH PICTURE SYMBOL '#'.

WORKING STORAGE SECTION.
 01 USPRICE PIC \$99.99.
 01 UKPRICE PIC £99.99.
 01 ECPRIICE PIC #99.99.

LENGTH must be specified when PICOUT specifies a COBOL picture containing a currency symbol that will be replaced by a currency sign of length greater than 1.

POS

specifies the location of a field. This operand specifies the individually addressable character location in a map at which the attribute byte that precedes the field is positioned.

number specifies the displacement (relative to zero) from the beginning of the map being defined.

(line,column)

specify lines and columns (relative to one) within the map being defined.

The location of data on the output medium is also dependent on DFHMDFI operands.

The first position of a field is reserved for an attribute byte. When supplying data for input mapping from non-3270 devices, the input data must allow space for this attribute byte. Input data must not start in column 1 but may start in column 2.

The POS operand always contains the location of the first position in a field, which is normally the attribute byte when communicating with the 3270. For the second and subsequent fields of a group, the POS operand points to an assumed attribute-byte position, ahead of the start of the data, even though no actual attribute byte is necessary. If the fields follow on immediately from one another, the POS operand should point to the last character position in the previous field in the group.

When a position number is specified that represents the last character position in the 3270, two special rules apply:

- ATTRIB=IC should not be coded. The cursor can be set to location zero by using the CURSOR option of a SEND MAP, SEND CONTROL, or SEND TEXT command.
- If the field is to be used in an output mapping operation with MAP=DATAONLY on the SEND MAP

command, an attribute byte for that field must be supplied in the symbolic map data structure by the application program.

PS specifies that programmed symbols are to be used. This overrides any PS operand set by the DFHMDFI macro or the DFHMDF macro.

BASE is the default and specifies that the base symbol set is to be used.

psid specifies a single EBCDIC character, or a hexadecimal code of the form X'nn', that identifies the set of programmed symbols to be used.

The PS operand is ignored unless the terminal supports programmed symbols, as indicated by PROGSYMBOLS(YES) on the RDO TYPETERM definition.

SOSI

indicates that the field may contain a mixture of EBCDIC and DBCS data. The DBCS subfields within an EBCDIC field are delimited by SO (shift out) and SI (shift in) characters. SO and SI both occupy a single screen position (normally displayed as a blank). They can be included in any non-DBCS field on output, if they are correctly paired. The terminal user can transmit them inbound if they are already present in the field, but can add them to an EBCDIC field only if the field has the SOSI attribute.

TRANSP

determines whether the background of an alphanumeric field is transparent or opaque, that is, whether an underlying (graphic) presentation space is visible between the characters.

VALIDN

specifies that validation is to be used on an 8775 terminal. This overrides any VALIDN operand on the DFHMDFI macro or the DFHMDF macro.

MUSTFILL specifies that the field must be filled completely with data. An attempt to move the cursor from the field before it has been filled, or to transmit data from an incomplete field, raises the INHIBIT INPUT condition

MUSTENTER

specifies that data must be entered into the field, though need not fill it. An attempt to move the cursor from an empty field raises the INHIBIT INPUT condition

TRIGGER specifies that this field is a trigger field. Trigger fields are discussed in the *CICS Application Programming Guide*.

The VALIDN operand is ignored unless the terminal supports validation, as indicated by VALIDATION(YES) option on the RDO TYPETERM definition.

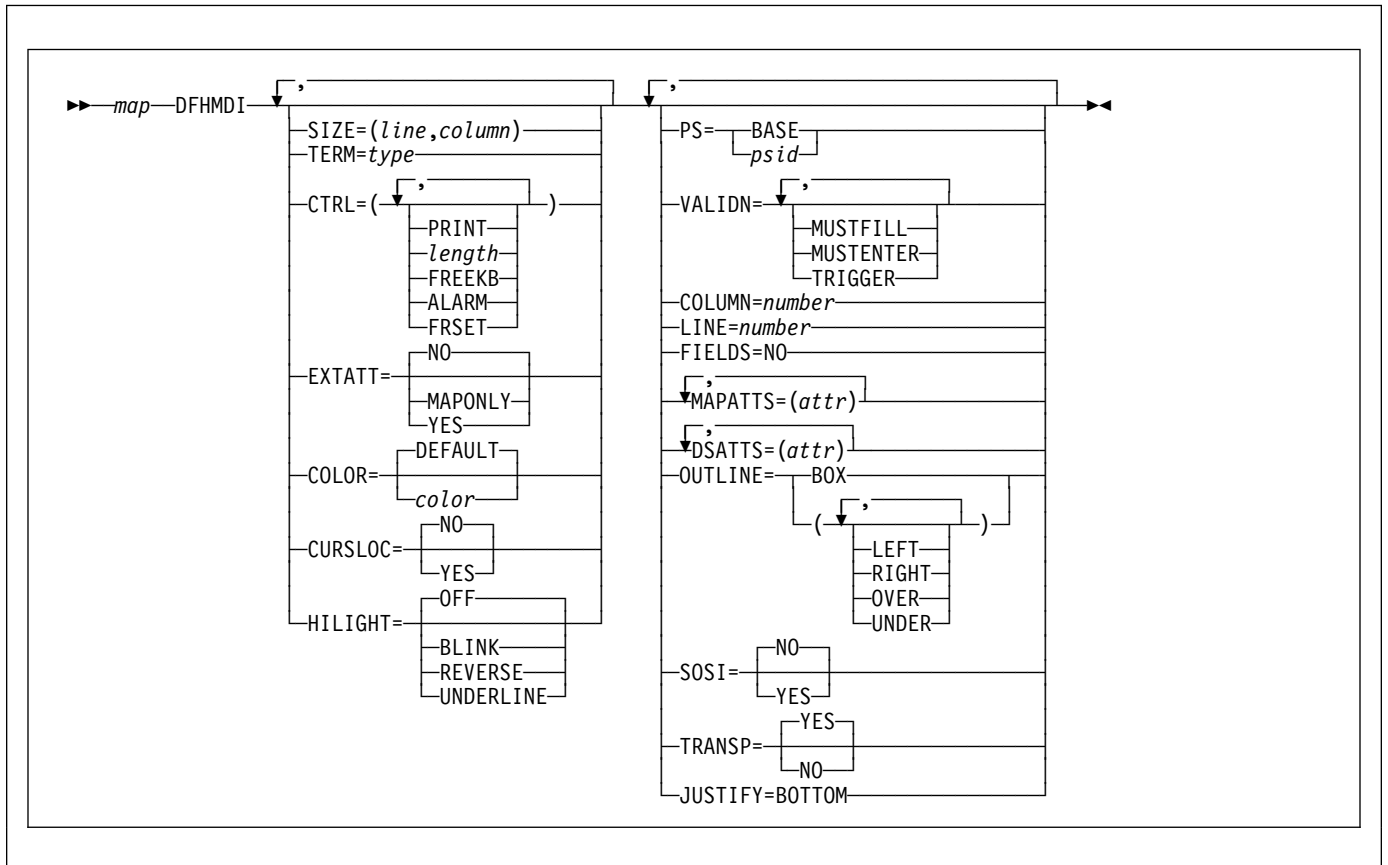
DFHMDF**XINIT**

see INITIAL, earlier in the list. Only one of GINIT, INITIAL, or XINIT may be specified.

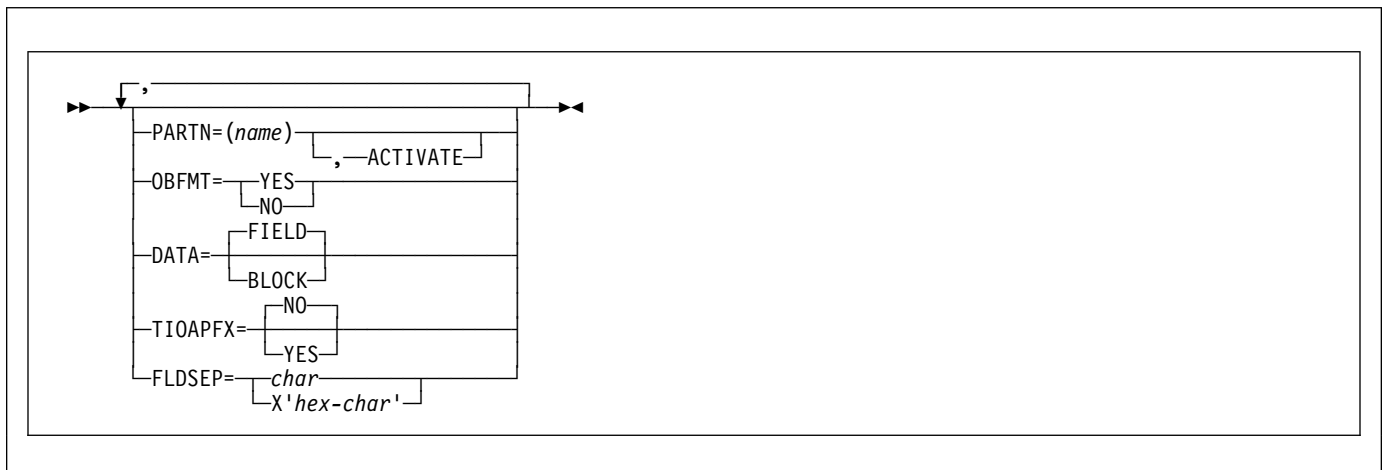
DFHMDI

Map definition.

Minimum BMS:

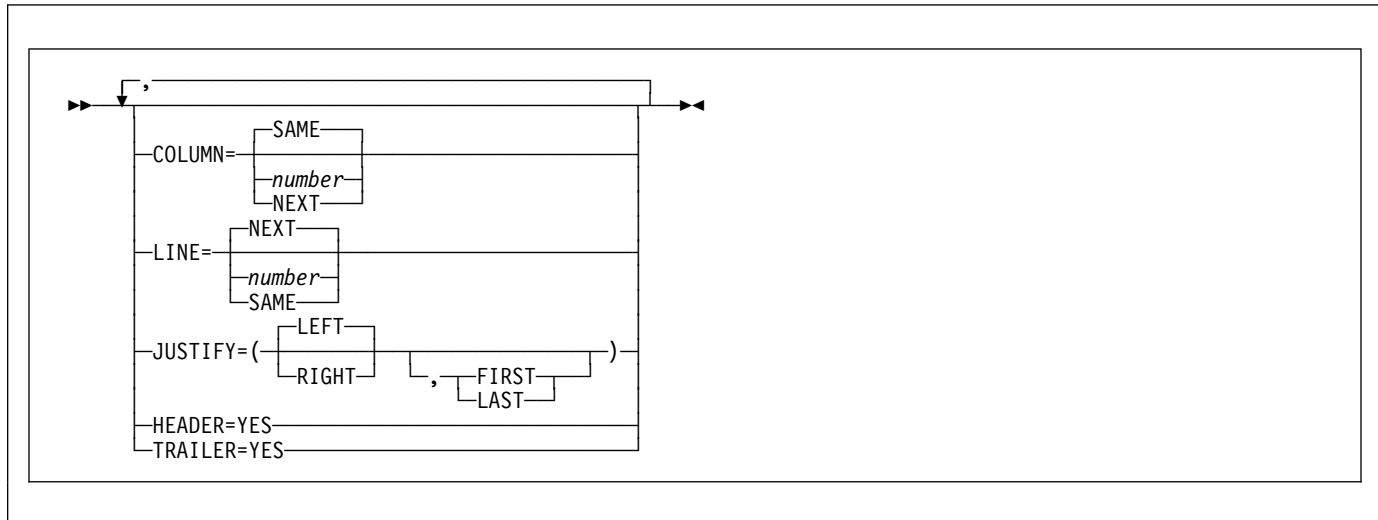


Standard BMS:



DFHMDI

Full BMS:



The DFHMDI macro defines a map within the mapset defined by a previous DFHMSD macro. A map contains zero or more fields.

“map” is the name (1–7 characters) of the map.

Note for COBOL users: If the maps are for use in a COBOL program, and STORAGE=AUTO has not been specified in the DFHMSD macro, they must be specified in descending size sequence. (Size refers to the generated 01-level data areas and not to the size of the map on the screen.) You should ensure that the names of maps, and names of fields within a mapset (or within multiple mapsets that are copied into one application program) are unique. Maps and mapsets must not have the same name.

Before CICS can load a physical map, you must define the physical map with an RDO MAPSET resource definition.

operands

COLOR

indicates the individual color, or the default color for the mapset (where applicable). This is overridden by the COLOR operand of the DFHMDF macro.

The valid colors are blue, red, pink, green, turquoise, yellow, and neutral.

The COLOR operand is ignored unless the terminal supports color, as indicated by the RDO TYPETERM definition option COLOR(YES).

COLUMN

specifies the column in a line at which the map is to be placed, that is, it establishes the left or right map margin. The JUSTIFY operand of the DFHMDI macro controls whether map and page margin selection and column counting are to be from the left or right side of the page. The columns between the specified map margin and the page margin are not available for subsequent use on the page for any lines included in the map.

NUMBER is the column from the left or right page margin where the left or right map margin is to be established.

NEXT indicates that the left or right map margin is to be placed in the next available column from the left or right on the current line.

SAME indicates that the left or right map margin is to be established in the same column as the last nonheader or nontrailer map used that specified COLUMN=number and the same JUSTIFY operands as this macro.

For input operations, the map is positioned at the extreme left-hand or right-hand side, depending on whether JUSTIFY=LEFT or JUSTIFY=RIGHT has been specified.

CTRL

defines characteristics of IBM 3270 terminals. Use of **any** of the control options in the SEND MAP command overrides **all** control options in the DFHMDI macro, which in turn overrides **all** control options in the DFHMSD macro.

If CTRL is used with cumulative BMS paging (that is, the ACCUM option is used on the BMS SEND MAP commands), it must be specified on the last (or only) map of a page, unless it is overridden by the ALARM, FREEKB and so on, options on the SEND MAP or accumulated SEND CONTROL command.

PRINT must be specified if the printer is to be started; if omitted, the data is sent to the printer buffer but is not printed. This operand is ignored if the mapset is used with 3270 displays without the Printer Adapter feature.

LENGTH indicates the line length on the printer; length can be specified as L40, L64, L80, or HONEOM. L40, L64, and L80 force a new

line after 40, 64, or 80 characters, respectively. HONEOM causes the default printer line length to be used. If this option is omitted, BMS sets the line length from the TCT page size.

FREEKB causes the keyboard to be unlocked after the map is written. If FREEKB is not specified, the keyboard remains locked; data entry from the keyboard is inhibited until this status is changed.

ALARM activates the 3270 audible alarm. For non-3270 VTAM terminals it sets the alarm flag in the FMH. (This feature is not supported by interactive and batch logical units.)

FRSET specifies that the modified data tags (MDTs) of all fields currently in the 3270 buffer are to be reset to an unmodified condition (that is, field reset) before map data is written to the buffer. This allows the DFHMDF macro with the ATTRB operand to control the final status of any fields written or rewritten in response to a BMS command.

Note: CTRL cannot be specified in the DFHMDI and DFHMDS macros in the same mapset.

CURSLOC

indicates that for all RECEIVE MAP operations using this map on 3270 terminals, BMS sets a flag in the application data structure element for the field where the cursor is located.

The flag may be tested by DFHBMCUR (see copybook DFHBMSCA in Appendix J, "BMS-related constants" on page 405).

To test the flag (COBOL example):

```
(DFHBMSCA)
...
02 DFHBMEOF PIC X VALUE X'80'.
02 DFHBMCUR PIC X VALUE X'02'.
02 DFHBMEC PIC X VALUE X'82'.
02 DFHBMFLG PIC X.
    88 DFHERASE VALUES ARE X'80', X'82'.
    88 DFHCURSR VALUES ARE X'02', X'82'.
MOVE FLD1F TO DFHBMFLG.
IF DFHERASE THEN ...
    ELSE ...
IF DFHCURSR THEN ...
    ELSE ...
```

Notes:

1. If CURSLOC=YES is specified for the MAP definitions, and there is no data for any field of the application data structure, but the cursor lies within a field known to the application data structure, BMS sets the cursor flag for the appropriate field, but the data for all fields in the application data structure is null, and the MAPFAIL condition does not occur. The unmapped data stream is not available to the

application program unless it is a RECEIVE DATA FROM request.

2. A valid CURSLOC definition in DFHMDI overrides the definition in DFHMDS.

DATA

specifies the format of the data.

FIELD specifies that the data is passed as contiguous fields, each field having the format:

LL	A	data field
----	---	------------

"LL" is two bytes specifying the length of the data as input from the terminal (ignored in output processing). "A" is a byte into which the programmer can place an attribute to override that specified in the map used to process this data (see copybook DFHBMSCA in Appendix J, "BMS-related constants" on page 405).

BLOCK

specifies that the data is passed as a continuous stream in the following format:

A	data field	space
---	------------	-------

This stream is processed as line segments of the length specified in the map used to process the data. The data is in the form in which it appears at the terminal; that is, it contains data fields and interspersed blanks corresponding to any spaces that are to appear between the fields on output. You cannot use DSATTS=YES if you specify DATA=BLOCK.

Block data is further discussed in the *CICS Application Programming Guide*.

DSATTS

specifies the attribute types to be included in the symbolic description map. These types can be one or more of the following: COLOR, HILIGHT, OUTLINE, PS, SOSI, TRANSP, and VALIDN. Any type included in DSATTS should also be included in MAPATTS.

EXTATT

this operand is supported for compatibility with previous releases. Each of the extended attributes can be defined individually. For new maps, the operands DSATTS and MAPATTS should be used instead.

NO is equivalent to specifying neither the DSATTS operand nor the MAPATTS operand.

YES is equivalent to:
 MAPATTS=(COLOR,HILIGHT,PS,VALIDN)
 DSATTS=(COLOR,HILIGHT,PS,VALIDN)

DFHMDI

MAPONLY is equivalent to:

MAPATTS=(COLOR,HILIGHT,PS,VALIDN)

FIELDS

specifies whether or not the map contains fields. If you specify FIELDS=NO, you create a null map that defines a "hole" in BMS's view of the screen. BMS cannot change the contents of such a hole after it has created it by sending a null map.

FLDSEP

specifies the field separator sequence (1–4 characters) for input from non-3270 devices. Input from non-3270 devices can be entered as a single string of data with the field separator sequence delimiting fields. The data between the field separators is moved to the input fields in the map in order.

HEADER

allows the map to be used during page building without terminating the OVERFLOW condition. This operand may be specified for more than one map in a mapset.

HILIGHT

specifies the default highlighting attribute for all fields in all maps in a mapset. This is overridden by the HILIGHT operand of the DFHMDF.

OFF is the default and indicates that no highlighting is used.

BLINK specifies that the field must blink.

REVERSE specifies that the character or field is displayed in reverse video, for example, on a 3278, black characters on a green background.

UNDERLINE

specifies that a field is underlined.

The HILIGHT operand is ignored unless the terminal supports highlighting, as indicated by HILIGHT(YES) on the RDO TYPETERM definition,

JUSTIFY

specifies the position of the map on the page.

LEFT specifies that the map is to be positioned starting at the specified column from the left margin on the specified line.

RIGHT specifies that the map is to be positioned starting at the specified column from the right margin on the specified line.

FIRST specifies that the map is to be positioned as the first map on a new page. Any partially formatted page from preceding BMS commands is considered to be complete. This operand can be specified for only one map per page.

LAST indicates that the map is to be positioned at the bottom of the current page. This operand can be specified for multiple maps

to be placed on one page. However, maps other than the first map for which it is specified must be able to be positioned horizontally without requiring that more lines be used.

BOTTOM

for a SEND MAP ACCUM command has the same effect as LAST, above. For a SEND MAP command (without ACCUM) and a RECEIVE MAP command, JUSTIFY=BOTTOM positions the map at the bottom of the screen if the number of lines in the map is specified in the SIZE operand. No account is taken of trailer maps in the mapset. JUSTIFY=BOTTOM is equivalent to specifying

LINE=(screendepth-mapdepth+1)

on the map definition, but it allows the same map to be used for different screen sizes. JUSTIFY=BOTTOM is ignored if the number of lines is not also specified. If JUSTIFY=BOTTOM and LINE are both specified, the value specified in LINE is ignored.

LEFT and RIGHT are mutually exclusive, as are FIRST and LAST. If neither FIRST nor LAST is specified, the data is mapped at the next available position as determined by other parameters of the map definition and the current mapping operation. FIRST or LAST is ignored unless ACCUM is specified on SEND MAP commands; otherwise only one map is placed on each page.

Note: If a field is initialized by an output map or contains data from any other source, data that is keyed as input overwrites only the equivalent length of the existing data; surplus existing data remains in the field and could cause unexpected interpretation of the new data.

LANG

supported for compatibility with previous releases.

LINE

specifies the starting line on a page in which data for a map is to be formatted.

NUMBER is a value in the range 1–240, specifying a starting line number. A request to map, on a line and column, data that has been formatted in response to a preceding BMS command, causes the current page to be treated as though complete. The new data is formatted at the requested line and column on a new page.

NEXT

specifies that formatting of data is to begin on the next available completely empty line. If LINE=NEXT is specified in the DFHMDI macro, it is ignored for input operations and LINE=1 is assumed.

SAME specifies that formatting of data is to begin on the same line as that used for a preceding BMS command. If COLUMN=NEXT is specified, it is ignored for input operations and COLUMN=1 is assumed. If the data does not fit on the same line, it is placed on the next available line that is completely empty.

MAPATTS

specifies the attribute types to be included in the physical map. These types can be one or more of the following: COLOR, HIGHLIGHT, OUTLINE, PS, SOSI, TRANSP, and VALIDN. This list must include all the attribute types to be specified for individual fields in the map (DFHMDF macro).

Where possible these values are deduced from operands already specified in the DFHMDI and DFHMDS macros. For example, if COLOR=BLUE has been specified, MAPATTS=COLOR is assumed.

MODE

kept for compatibility with previous releases.

OBFMT

specifies whether outboard formatting is to be used. This operand is available only for 3650 logical units, or for an 8100 series processor running DPS Release 2 and defined to CICS as an LUTYPE2 logical unit. For more information, see the *CICS Application Programming Guide*.

The OBFMT operand overrides the OBFMT operand on the DFHMDS macro.

YES specifies that this map definition can be used in outboard formatting.

NO specifies that this map definition cannot be used in outboard formatting.

OUTLINE

allows lines to be included above, below, to the left, or to the right of a field. You can use these lines in any combination to construct boxes around fields or groups of fields.

PARTN

specifies the default partition to be associated with maps in this mapset. If the ACTIVATE option is specified, the specified partition is also activated when maps in this mapset are output to a terminal that supports partitions.

This option overrides the PARTN option of the DFHMDS macro and is overridden by any OUTPARTN or ACTPARTN option on the SEND MAP command, or the INPARTN option on a RECEIVE MAP command.

The PARTN option is ignored if the target terminal does not support partitions, or if there is no partition set associated with the transaction.

PS specifies that programmed symbols are to be used. This overrides the PS operand of the DFHMDS macro and is overridden by the PS operand of the DFHMDF macro.

BASE specifies that the base symbol set is to be used.

psid specifies a single EBCDIC character, or a hexadecimal code of the form X'nn', that identifies the set of programmed symbols to be used.

The PS operand is ignored unless the terminal supports programmed symbols, as indicated by the PROGSYMBOLS(YES) on the RDO TYPETERM definition.

SIZE

specifies the size of a map.

line is a value in the range 1–240, specifying the depth of a map as a number of lines.

column is a value in the range 1–240, specifying the width of a map as a number of columns.

This operand is required in the following cases:

- An associated DFHMDF macro with the POS operand is used.
- The map is to be referred to in a SEND MAP command with the ACCUM option.
- The map is to be used when referring to input data from other than a 3270 terminal in a RECEIVE MAP command.

SOSI

indicates that the field may contain a mixture of EBCDIC and DBCS data. The DBCS subfields within an EBCDIC field are delimited by SO (shift out) and SI (shift in) characters. SO and SI both occupy a single screen position (normally displayed as a blank). They can be included in any non-DBCS field on output, if they are correctly paired. The terminal user can transmit them inbound if they are already present in the field, but can add them to an EBCDIC field only if the field has the SOSI attribute.

TIOAPFX

specifies whether BMS should include a filler in the symbolic description maps to allow for the unused TIOA prefix. This operand overrides the TIOAPFX operand specified for the DFHMDS macro.

YES specifies that the filler should be included in the symbolic description maps. If TIOAPFX=YES is specified, all maps within the mapset have the filler. TIOAPFX=YES should **always** be used for command-level application programs.

NO is the default and specifies that the filler is not to be included.

DFHMDI

TRAILER

allows the map to be used during page building without terminating the OVERFLOW condition. This operand may be specified for more than one map in a mapset. If a trailer map is used other than in the overflow environment, the space normally reserved for overflow trailer maps is not reserved while mapping the trailer map.

TRANSP

determines whether the background of an alphanumeric field is transparent or opaque, that is, whether an underlying (graphic) presentation space is visible between the characters.

TYPE

kept for compatibility with previous releases.

VALIDN

specifies that validation is to be used on an 8775 terminal. This is overridden by the VALIDN operand of the DFHMDF macro, and overrides the VALIDN operand of the DFHMDS macro.

MUSTFILL specifies that the field must be filled completely with data. An attempt to move the cursor from the field before it has been filled, or to transmit data from an incomplete field, raises the INHIBIT INPUT condition.

MUSTENTER

specifies that data must be entered into the field, though need not fill it. An attempt to move the cursor from an empty field raises the INHIBIT INPUT condition.

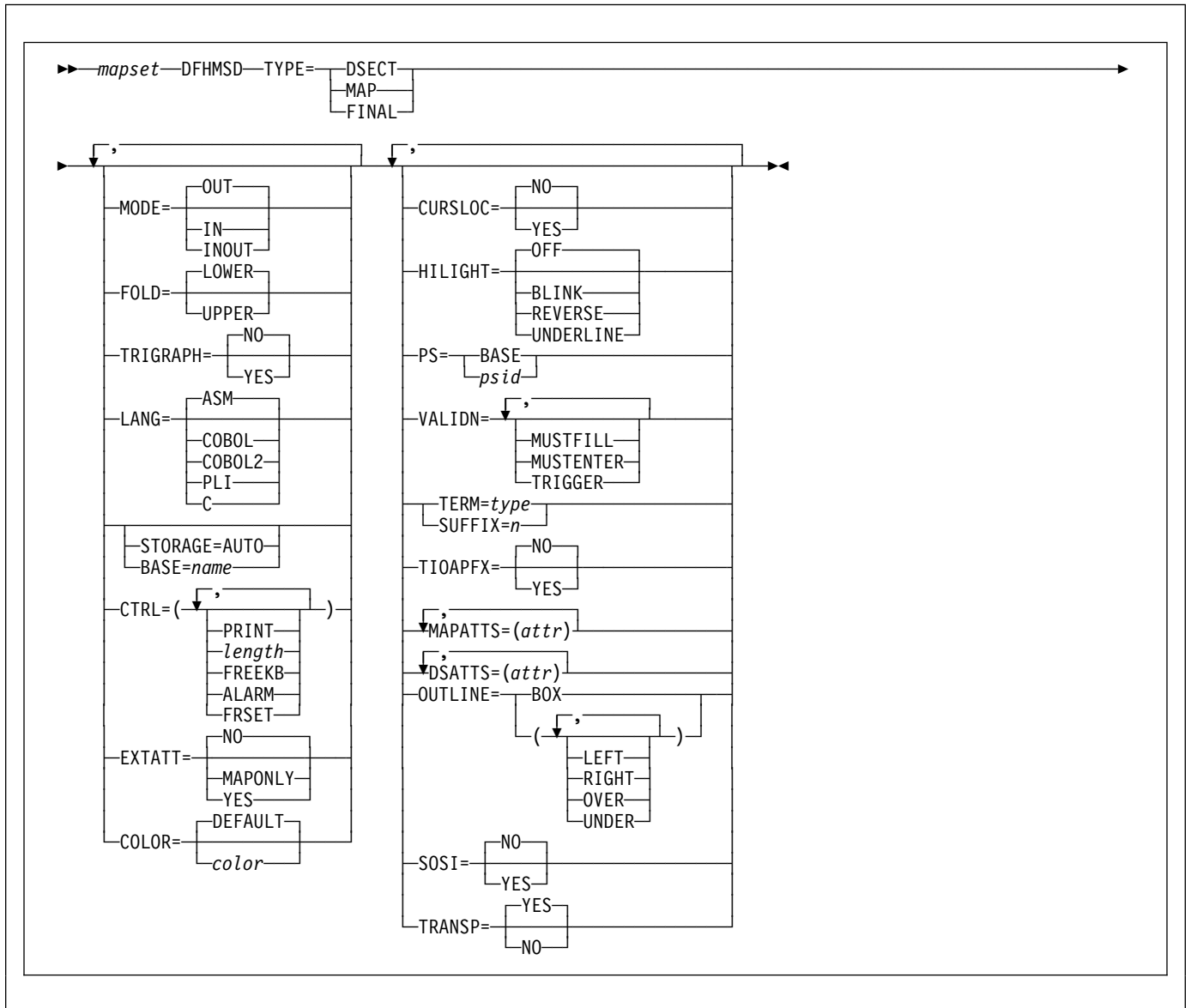
TRIGGER specifies that this field is a trigger field. Trigger fields are discussed in the *CICS Application Programming Guide*.

The VALIDN operand is ignored unless the terminal supports validation, as indicated by VALIDATION(YES) on the RDO TYPETERM definition.

DFHMSD

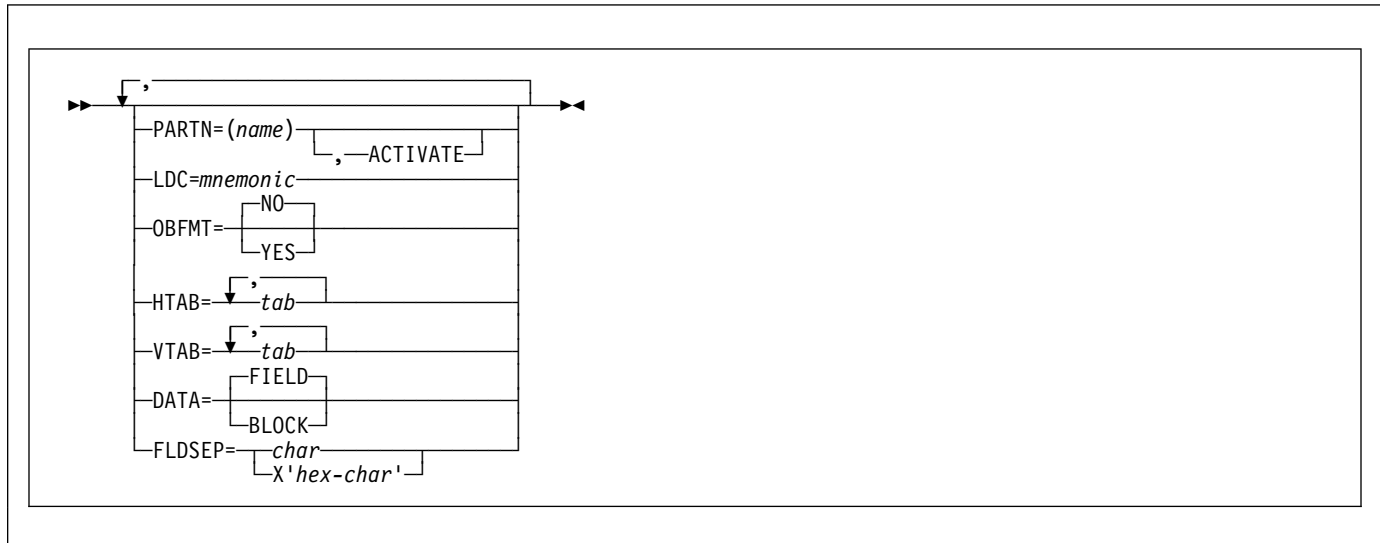
Mapset definition.

Minimum BMS:



DFHMSD

Standard BMS:



A DFHMSD macro defines a mapset; it begins:

```
DFHMSD TYPE=MAP (or TYPE=DSECT)
```

and ends:

```
DFHMSD TYPE=FINAL
```

“mapset” is the name (1–7 characters) of the mapset.

A DFHMSD macro contains one or more map definition macros, each of which contains one or more field definition macros.

You should ensure that the names of maps, and names of fields within a mapset (or within multiple mapsets that are copied into one application program) are unique. Maps and mapsets must not have the same name.

Before CICS can load a physical map, you must define the physical map with an RDO MAPSET resource definition.

You assemble a BMS mapset definition to generate either a symbolic description map or a physical map. The *CICS System Definition Guide* tells you how to assemble and catalog the maps.

operands

BASE

specifies that the same storage base is used for the symbolic description maps from more than one mapset. The same name is specified for each mapset that is to share the same storage base. Because all mapsets with the same base describe the same storage, data related to a previously used mapset may be overwritten when a

new mapset is used. Different maps within the same mapset also overlay one another.

This operand is not valid for assembler-language programs, and cannot be used when STORAGE=AUTO has been specified.

COLOR

indicates the individual color, or the default color for the mapset (where applicable). This is overridden by the COLOR operand of the DFHMDI macro, which is in turn overridden by the COLOR operand of the DFHMDF macro.

The valid colors are blue, red, pink, green, turquoise, yellow, and neutral.

The COLOR operand is ignored unless the terminal supports color, as indicated by the RDO TYPETERM definition COLOR(YES) option.

CTRL

defines characteristics of IBM 3270 terminals. Use of **any** of the control options in the SEND MAP command overrides **all** control options in the DFHMDI macro, which in turn overrides **all** control options in the DFHMSD macro.

If CTRL is used with cumulative BMS paging (that is, the ACCUM option is used on the BMS SEND MAP commands), it must be specified on the last (or only) map of a page, unless it is overridden by the ALARM, FREEKB and so on, options on the SEND MAP or accumulated SEND CONTROL command.

PRINT

must be specified if the printer is to be started; if omitted, the data is sent to the printer buffer but is not printed. This operand is ignored if the mapset is used with 3270 displays without the Printer Adapter feature.

- LENGTH** indicates the line length on the printer; length can be specified as L40, L64, L80, or HONEOM. L40, L64, and L80 force a new line after 40, 64, or 80 characters, respectively. HONEOM causes the default printer line length to be used. If this option is omitted, BMS sets the line length from the TCT page size.
- FREEKB** causes the keyboard to be unlocked after the map is written. If FREEKB is not specified, the keyboard remains locked; data entry from the keyboard is inhibited until this status is changed.
- ALARM** activates the 3270 audible alarm. For non-3270 VTAM terminals, it sets the alarm flag in the FMH. (This feature is not supported by interactive and batch logical units.)
- FRSET** specifies that the modified data tags (MDTs) of all fields currently in the 3270 buffer are to be reset to a not-modified condition (that is, field reset) before map data is written to the buffer. This allows the DFHMDF macro with the ATTRB operand to control the final status of any fields written or rewritten in response to a BMS command.

Note: CTRL cannot be specified in the DFHMDI and DFHMSD macros in the same mapset.

CURSLOC

indicates that for all RECEIVE MAP operations using this map on 3270 terminals, BMS sets a flag in the application data structure element for the field where the cursor is located.

The flag may be tested by DFHBMCUR (see copybook DFHBMSCA in Appendix J, "BMS-related constants" on page 405).

To test the flag (COBOL example):

(DFHBMSCA)

```

...
02 DFHMEOF    PIC X VALUE X'80'.
02 DFHMCUR    PIC X VALUE X'02'.
02 DFHMEC     PIC X VALUE X'82'.
02 DFHMFLG    PIC X.
    88 DFHERASE    VALUES ARE X'80', X'82'.
    88 DFHCURSR   VALUES ARE X'02', X'82'.
MOVE FLD1F TO DFHMFLG.
IF DFHERASE THEN ...
    ELSE ...
IF DFHCURSR THEN ...
    ELSE ...

```

Notes:

1. If CURSLOC=YES is specified for the MAP definitions, and there is no data for any field of the application data structure, but the cursor lies within a field known to the application data structure, BMS sets the cursor flag for the appropriate field, but the data for all fields in the application data structure is null, and the MAPFAIL condition does not occur. The unmapped data stream is not available to the application program unless it is a RECEIVE DATA FROM request.
2. A valid CURSLOC definition in DFHMDI overrides the definition in DFHMSD.

DATA

specifies the format of the data.

FIELD specifies that the data is passed as contiguous fields, each field having the format:

LL	A	data field
----	---	------------

"LL" is two bytes specifying the length of the data as input from the terminal (these two bytes are ignored in output processing). "A" is a byte into which the programmer can place an attribute to override that specified in the map used to process this data (see copybook DFHBMSCA in Appendix J, "BMS-related constants" on page 405).

BLOCK

specifies that the data is passed as a continuous stream in the following format:

A	data field	space
---	------------	-------

This stream is processed as line segments of the length specified in the map used to process the data. The data is in the form that it appears on the terminal; that is, it contains data fields and interspersed blanks corresponding to any spaces that are to appear between the fields on output. You cannot use DSATTS=YES, if you specify DATA=BLOCK.

Block data is further discussed in the *CICS Application Programming Guide*.

DSATTS

specifies the attribute types to be included in the symbolic description map. These types can be one or more of the following: COLOR, HILIGHT, OUTLINE, PS, SOSI, TRANSP, and VALIDN. Any type included in DSATTS should also be included in MAPATTS.

EXTATT

this operand is supported for compatibility with previous releases. Each of the extended attributes can be

DFHMSD

defined individually. For new maps, the operands DSATTS and MAPATTS should be used instead.

NO is equivalent to specifying neither the DSATTS operand nor the MAPATTS operand.

YES is equivalent to:
MAPATTS=(COLOR,HILIGHT,PS,VALIDN)
DSATTS=(COLOR,HILIGHT,PS,VALIDN)

MAPONLY is equivalent to:
MAPATTS=(COLOR,HILIGHT,PS,VALIDN)

FLDSEP

specifies the field separator sequence (1–4 characters) for input from non-3270 devices. Input from non-3270 devices can be entered as a single string of data with the field separator sequence delimiting fields. The data between the field separators is moved to the input fields in the map in order.

FOLD

specifies whether to generate lowercase or uppercase characters in C language programs.

FOLD is only available for programs written in C.

HILIGHT

specifies the default highlighting attribute for all fields in all maps in a mapset. This is overridden by the HILIGHT operand of the DFHMDI, which is in turn overridden by the HILIGHT operand of the DFHMDF.

OFF is the default and indicates that no highlighting is used.

BLINK specifies that the field must blink.

REVERSE specifies that the character or field is displayed in reverse video, for example, on a 3278, black characters on a green background.

UNDERLINE specifies that a field is underlined.

The HILIGHT operand is ignored unless the terminal supports highlighting, as indicated by HILIGHT(YES) on the RDO TYPETERM definition.

HTAB

specifies one or more tab positions for use with interactive and batch logical units and SCS printers with horizontal forms control.

LANG

specifies the source language of the application programs into which the symbolic description maps in the mapset are copied. This option need only be coded for DFHMSD TYPE=DSECT. If a mapset is to be used by more than one program, and the programs are not all written in the same source language, a separate version of the mapset must be defined for each programming language.

LDC

specifies the code to be used by CICS to determine the logical device mnemonic to be used for a BMS output operation. If no LDC operand has been specified on any previous BMS output in the logical message, this LDC will be transmitted in the function management header to the logical unit. This operand is used only for VTAM-supported 3600 terminals, and batch logical units.

MAPATTS

specifies the attribute types to be included in the physical map. These types can be one or more of the following: COLOR, HILIGHT, OUTLINE, PS, SOSI, TRANSP, and VALIDN. This list must include all the attribute types to be specified for individual fields in the map (DFHMDF macro).

Where possible these values are deduced from operands already specified in the DFHMDI and DFHMSD macros. For example, if COLOR=BLUE has been specified, MAPATTS=COLOR is assumed.

MODE

specifies whether the mapset is to be used for input, output, or both.

OBFMT

specifies whether outboard formatting is to be used. This operand is available only for 3650 logical units, or for an 8100 series processor running DPS Release 2 and defined to CICS as an LUTYPE2 logical unit. For more information, see the *CICS Application Programming Guide*.

The OBFMT operand on DFHMSD is overridden by the OBFMT operand on DFHMDI.

YES specifies that all maps within this mapset can be used in outboard formatting, except those for which OBFMT=NO is specified in the DFHMDI macro.

NO specifies that no maps within this mapset can be used in outboard formatting, except those for which OBFMT=YES is specified in DFHMDI.

OUTLINE

allows lines to be included above, below, to the left, or to the right of a field. You can use these lines in any combination to construct boxes around fields or groups of fields.

PARTN

specifies the default partition to be associated with maps in this mapset. If the ACTIVATE option is specified, the specified partition is also activated when maps in this mapset are output to a terminal that supports partitions. This option is overridden by the PARTN operand of the DFHMDI macro, which is in turn overridden by any OUTPARTN or ACTPARTN option on the SEND MAP command, or the INPARTN option on a RECEIVE MAP command.

The PARTN operand is ignored if the target terminal does not support partitions, or if there is no partition set associated with the transaction.

PS specifies that programmed symbols are to be used. This is overridden by the PS operand of the DFHMDI macro, which is in turn overridden by the PS operand of the DFHMDF macro.

BASE specifies that the base symbol set is to be used.

psid specifies a single EBCDIC character, or a hexadecimal code of the form X'nn', that identifies the set of programmed symbols to be used.

The PS operand is ignored unless the terminal supports programmed symbols, as indicated by PROGSYMBOLS(YES) on the RDO TYPETERM definition.

SOSI

indicates that the field may contain a mixture of EBCDIC and DBCS data. The DBCS subfields within an EBCDIC field are delimited by SO (shift out) and SI (shift in) characters. SO and SI both occupy a single screen position (normally displayed as a blank). They can be included in any non-DBCS field on output provided they are correctly paired. The terminal user can transmit them inbound if they are already present in the field, but can add them to an EBCDIC field only if the field has the SOSI attribute.

STORAGE

The meaning of this operand depends upon the language in which application programs are written, as follows:

For a **COBOL** program, STORAGE=AUTO specifies that the symbolic description maps in the mapset are to occupy separate (that is, not redefined) areas of storage. This operand is used when the symbolic description maps are copied into the working-storage section and the storage for the separate maps in the mapset is to be used concurrently.

For a **C** program, STORAGE=AUTO specifies that the symbolic description maps are to be defined as having the automatic storage class. If STORAGE=AUTO is not specified, they are declared as pointers. You cannot specify both BASE=name and STORAGE=AUTO for the same mapset. If STORAGE=AUTO is specified and TIOAPFX is not, TIOAPFX=YES is assumed.

For a **PL/I** program, STORAGE=AUTO specifies that the symbolic description maps are to be declared as having the AUTOMATIC storage class. If STORAGE=AUTO is not specified, they are declared as BASED. You cannot specify both BASE=name and STORAGE=AUTO for the same mapset. If STORAGE=AUTO is specified and TIOAPFX is not, TIOAPFX=YES is assumed.

For an **assembler-language** program, STORAGE=AUTO specifies that individual maps within a

mapset are to occupy separate areas of storage instead of overlaying one another.

SUFFIX

specifies a 1-character, user-defined, device-dependent suffix for this mapset, as an alternative to a suffix generated by the TERM operand. The suffix specified by this operand should match the value of a transaction defined with ALTSUFFIX, or ALTSFX in the terminal control table TYPE=TERMINAL. Use a numeric value to avoid conflict with suffixes generated by the TERM operand.

TERM

specifies the type of terminal or logical unit (LU) associated with the mapset. If no terminal type or LU is specified, 3270 is assumed. The terminal types and LUs you can specify, together with their generated suffixes, are shown in Table 31.

Table 31. BMS terminal types

TYPE	Suffix
CRLP ¹	A
TAPE	B
DISK	C
TWX	D
1050	E
2740	F
2741	G
2770	I
2780	J
3780	K
3270-1 (40-column)	L
3270-2 (80-column)	M
INTLU/3767/3770I/SCS ²	P
2980	Q
2980-4	R
3270 ³	blank
3601	U
3653 ⁴	V
3650UP ⁵	W
3650/3270 ⁶	X
BCHLU/3770B ⁷	Y
ALL (all the above)	blank
Notes:	
1. Card-reader-in/line-printer-out.	
2. All interactive LUs including 3790 full-function LU and SCS printer LUs (3270 and 3790).	
3. Default if TERM omitted. Same as ALL; used when no need to distinguish between models.	
4. Plus host-conv (3653) LU.	
5. Plus interpreter LU.	
6. Plus host-conv (3270) LU.	
7. Plus all batch and BDI LUs.	

In addition, you should note the following:

If ALL is specified, ensure that device dependent characters are not included in the mapset and that format characteristics such as page size are suitable for

DFHMSD

all input/output operations (and all terminals) in which the mapset is applied. For example, some terminals are limited to 480 bytes, others to 1920 bytes; the 3604 is limited to six lines of 40 characters each. Within these guidelines, use of ALL can offer important advantages. Because an assembly run is required for each map generation, the use of ALL, indicating that one map is to be used for more than one terminal, can result in significant time and storage savings.

However, better run-time performance for maps used by single terminal types is achieved if the terminal type (rather than ALL) is specified. Alternatively, BMS support for device-dependent mapsets can be bypassed by specifying NODDS in the BMS operand of the system initialization parameters. For more information, see the *CICS System Definition Guide* manual.

TIOAPFX

specifies whether BMS should include a filler in the symbolic description maps to allow for the unused TIOA prefix.

YES specifies that the filler should be included in the symbolic description maps. If TIOAPFX=YES is specified, all maps within the mapset have the filler, except when TIOAPFX=NO is specified on the DFHMDI macro. TIOAPFX=YES should **always** be used for command-level application programs.

NO is the default and specifies that the filler is not to be included. The filler may still be included for a map if TIOAPFX=YES is specified on DFHMDI.

TRANSP

determines whether the background of an alphanumeric field is transparent or opaque, that is, whether an underlying (graphic) presentation space is visible between the characters.

TRIGRAPH

specifies trigraph sequences to be used in REPLACE WITH EITHER CLANG or CCOMP language symbolic description maps.

When TRIGRAPH=YES, trigraph sequences are produced:

```
{ prints as ??<
} prints as ??>
[ prints as ??(
] prints as ??)
```

This option is only available for programs written in C.

TYPE

specifies the type of map to be generated using the definition. Both types of map must be generated before the mapset can be used by an application program. If aligned symbolic description maps are required, you should ensure that you specify SYSPARM=ADSECT and

SYSPARM=AMAP when you assemble the symbolic and physical maps respectively.

DSECT specifies that a symbolic description map is to be generated. Symbolic description maps must be copied into the source program before it is translated and compiled.

MAP specifies that a physical map is to be generated. Physical maps must be assembled or compiled, link-edited, and cataloged in the CICS program library before an application program can use them.

If both map and DSECT are to be generated in the same job, the SYSPARM option can be used in the assembler job execution step, as described in the *CICS System Definition Guide*.

VALIDN

specifies that validation is to be used on an 8775 terminal. This is overridden by the VALIDN operand of the DFHMDI macro, which is in turn overridden by the VALIDN operand of the DFHMDF macro.

MUSTFILL specifies that the field must be filled completely with data. An attempt to move the cursor from the field before it has been filled, or to transmit data from an incomplete field, raises the INHIBIT INPUT condition.

MUSTENTER

specifies that data must be entered into the field, though need not fill it. An attempt to move the cursor from an empty field raises the INHIBIT INPUT condition.

TRIGGER specifies that this field is a trigger field. Trigger fields are discussed in the *CICS Application Programming Guide*.

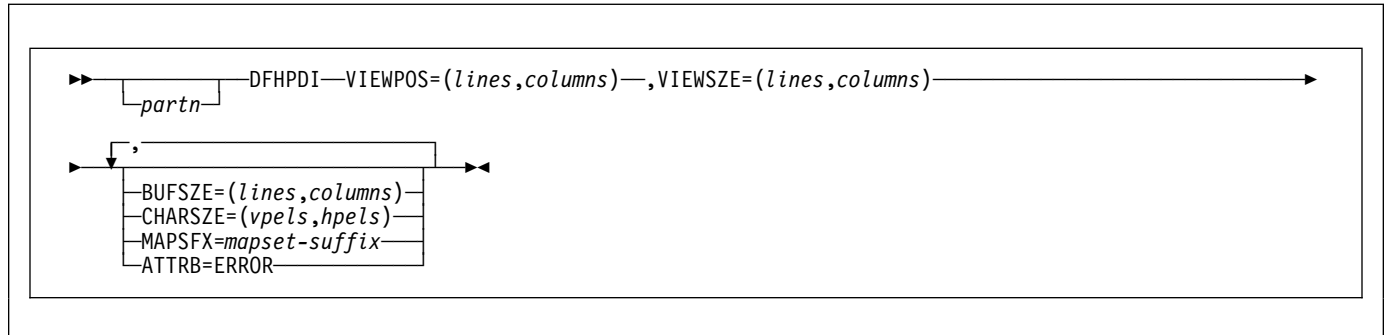
The VALIDN operand is ignored unless the terminal supports validation, as indicated by VALIDATION(YES) in the RDO TYPETERM definition.

VTAB

specifies one or more tab positions for use with interactive and batch logical units and SCS printers having vertical forms control.

DFHPDI

Partition definition.



A partition set contains one or more partitions. Each partition is defined by coding a partition definition macro.

“partn” is a partition name (1–2 characters). It allows you to refer to the partition in your application programs.

Every partition in a partition set must have a different name. Only the error partition can be unnamed (see ATTRB=ERROR operand).

Partitions are defined by coding the macros DFHPSD (partition set definition) and DFHPDI (partition definition). Each partition definition must be part of a partition set definition.

operands

ATTRB

specifies that error messages are to be directed to this partition whenever possible. The partition is cleared before an error message is displayed. The RDO TYPETERM option ERRHILIGHT is honored, but the LASTLINE option is ignored.

BUFSIZE(lines,columns)

specifies the size of the presentation space for the partition. Device limitations mean that the “columns” value must be equal to the “columns” value specified by the VIEWSIZE operand. The “lines” value can be greater than or, by default, equal to the value specified by the VIEWSIZE operand. A greater lines value implies that the target terminal supports vertical scrolling.

CHARSIZE(vpels, hpels)

specifies the size of the character cell to be reserved for each character displayed in a partition. You specify the size as numbers of vertical picture elements (vpels) and numbers of horizontal picture elements (hpels). You can specify this operand on either the DFHPSD macro only, or on both the DFHPSD and DFHPDI macros. The values specified in the DFHPSD become the defaults for all partitions in the partition set. You can override these

defaults for individual partitions by coding CHARSIZE in the DFHPDI macro.

MAPSFX(mapset-suffix)

specifies the partition’s 1-character mapset suffix. BMS uses the suffix to select mapset versions in the same way as for the RDO TYPETERM option ALTSUFFIX. If this operand is omitted, a suffix L is assumed if the “columns” value of the BUFSIZE operand is less than or equal to 40; otherwise M is assumed.

VIEWPOS(lines,columns)

specifies the position of the top left-hand corner of this partition’s viewport. You specify the position in numbers of lines and numbers of columns.

The DFHPDI macro checks that viewports do not overlap. If you have coded the RDO TYPETERM ALTSCREEN option, or the ALTSCRN operand of the DFHPSD macro, DFHPDI also checks that all viewports fit within the usable area of the terminal screen.

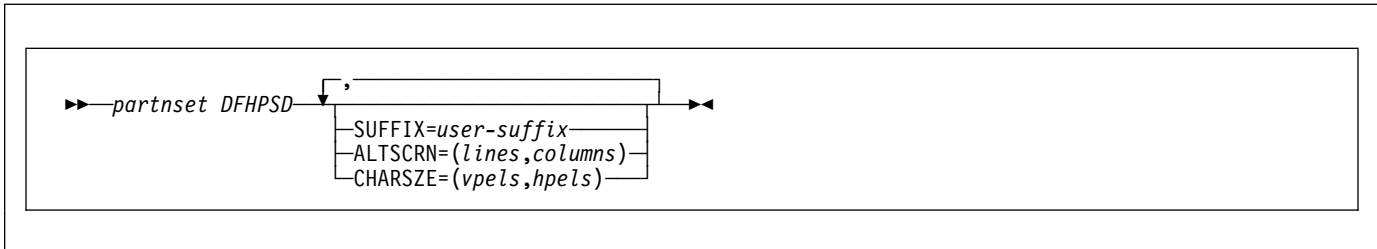
Note: The information given here on positioning viewports is necessarily brief. For more information you should consult the component description for the device you are using.

VIEWSIZE(lines,columns)

specifies the size, in lines and columns, of the partition’s viewport. The DFHPDI macro checks that viewports do not overlap. If you code the RDO TYPETERM ALTSCREEN option, or the ALTSCRN operand of the DFHPSD macro partition set definition macro, DFHPDI checks that the partitions all fit within the usable area of the display screen.

DFHPSD

Partition set definition.



Each partition set definition contains a single DFHPSD macro followed by one or more DFHPDI macros, and ending with a DFHPSD TYPE=FINAL partition set definition macro.

“partnset” is a partition set name (1–6 characters).

Partitions are defined by coding the macros DFHPSD (partition set definition) and DFHPDI (partition definition). Each partition definition must be part of a partition set definition.

ending DFHPSD

```
[partnset] DFHPSD TYPE=FINAL
```

The PARTNSET name (if specified) must match that specified on the DFHPSD macro that started the partition set definition.

operands

ALTSCRN(lines,columns)

specifies the size, in characters, of the usable area of the target terminal. This is normally the same as the RDO TYPETERM option ALTSCREEN. You use ALTSCRN to ensure that the viewports of partitions within a partition set fit into the usable area of the screen.

CHARSIZE(vpels,hpels)

specifies the size of the character cell to be reserved for each character displayed in a partition. You specify the size as numbers of vertical picture elements (vpels) and numbers of horizontal picture elements (hpels). You can specify this operand on either the DFHPSD macro only, or on both the DFHPSD and DFHPDI macros. The values specified in this operand become the defaults for all partitions in the partition set. You can override this default for individual partitions by coding CHARSIZE in the DFHPDI macro.

SUFFIX(user-suffix)

specifies a 1-character user suffix for this version of the partition set. It allows different versions of a partition set

to be associated with different terminals. When the partition set is to be loaded, CICS looks for a version whose suffix matches the RDO TYPETERM option ALTSUFFIX. If it cannot find the correct partition set version, it loads a version with a default suffix (M or L). If it cannot find a suffixed version either, it loads an unsuffixed one. If it cannot find this, it abends with APCT.

Bibliography

CICS Transaction Server for VSE/ESA Release 1 library

Evaluation and planning	
<i>Release Guide</i>	GC33-1645
<i>Migration Guide</i>	GC33-1646
<i>Report Controller Planning Guide</i>	SC33-1941
General	
<i>Master Index</i>	SC33-1648
<i>Trace Entries</i>	SX33-6108
<i>User's Handbook</i>	SX33-6101
<i>Glossary (softcopy only)</i>	GC33-1649
Administration	
<i>System Definition Guide</i>	SC33-1651
<i>Customization Guide</i>	SC33-1652
<i>Resource Definition Guide</i>	SC33-1653
<i>Operations and Utilities Guide</i>	SC33-1654
<i>CICS-Supplied Transactions</i>	SC33-1655
Programming	
<i>Application Programming Guide</i>	SC33-1657
<i>Application Programming Reference</i>	SC33-1658
<i>Sample Applications Guide</i>	SC33-1713
<i>Application Migration Aid Guide</i>	SC33-1943
<i>System Programming Reference</i>	SC33-1659
<i>Distributed Transaction Programming Guide</i>	SC33-1661
<i>Front End Programming Interface User's Guide</i>	SC33-1662
Diagnosis	
<i>Problem Determination Guide</i>	GC33-1663
<i>Messages and Codes Vol 3 (softcopy only)</i>	SC33-6799
<i>Diagnosis Reference</i>	LY33-6085
<i>Data Areas</i>	LY33-6086
<i>Supplementary Data Areas</i>	LY33-6087
Communication	
<i>Intercommunication Guide</i>	SC33-1665
<i>CICS Family: Interproduct Communication</i>	SC33-0824
<i>CICS Family: Communicating from CICS on System/390</i>	SC33-1697
Special topics	
<i>Recovery and Restart Guide</i>	SC33-1666
<i>Performance Guide</i>	SC33-1667
<i>Shared Data Tables Guide</i>	SC33-1668
<i>Security Guide</i>	SC33-1942
<i>External CICS Interface</i>	SC33-1669
<i>XRF Guide</i>	SC33-1671
<i>Report Controller User's Guide</i>	SC34-5688
CICS Clients	
<i>CICS Clients: Administration</i>	SC33-1792
<i>CICS Universal Clients Version 3 for OS/2: Administration</i>	SC34-5450
<i>CICS Universal Clients Version 3 for Windows: Administration</i>	SC34-5449
<i>CICS Universal Clients Version 3 for AIX: Administration</i>	SC34-5348
<i>CICS Universal Clients Version 3 for Solaris: Administration</i>	SC34-5451
<i>CICS Family: OO programming in C++ for CICS Clients</i>	SC33-1923
<i>CICS Family: OO programming in BASIC for CICS Clients</i>	SC33-1671
<i>CICS Family: Client/Server Programming</i>	SC33-1435
<i>CICS Transaction Gateway Version 3: Administration</i>	SC34-5448

Books from VSE/ESA 2.4 base program libraries

VSE/ESA Version 2 Release 4

Book title	Order number
Administration	SC33-6705
Diagnosis Tools	SC33-6614
Extended Addressability	SC33-6621
Guide for Solving Problems	SC33-6710
Guide to System Functions	SC33-6711
Installation	SC33-6704
Licensed Program Specification	GC33-6700
Messages and Codes Volume 1	SC33-6796
Messages and Codes Volume 2	SC33-6798
Messages and Codes Volume 3	SC33-6799
Networking Support	SC33-6708
Operation	SC33-6706
Planning	SC33-6703
Programming and Workstation Guide	SC33-6709
System Control Statements	SC33-6713
System Macro Reference	SC33-6716
System Macro User's Guide	SC33-6715
System Upgrade and Service	SC33-6702
System Utilities	SC33-6717
TCP/IP User's Guide	SC33-6601
Turbo Dispatcher Guide and Reference	SC33-6797
Unattended Node Support	SC33-6712

High-Level Assembler Language (HLASM)

Book title	Order number
General Information	GC26-8261
Installation and Customization Guide	SC26-8263
Language Reference	SC26-8265
Programmer's Guide	SC26-8264

Language Environment for VSE/ESA (LE/VSE)

Book title	Order number
C Run-Time Library Reference	SC33-6689
C Run-Time Programming Guide	SC33-6688
Concepts Guide	GC33-6680
Debug Tool for VSE/ESA Fact Sheet	GC26-8925
Debug Tool for VSE/ESA Installation and Customization Guide	SC26-8798
Debug Tool for VSE/ESA User's Guide and Reference	SC26-8797
Debugging Guide and Run-Time Messages	SC33-6681
Diagnosis Guide	SC26-8060
Fact Sheet	GC33-6679
Installation and Customization Guide	SC33-6682
LE/VSE Enhancements	SC33-6778
Licensed Program Specification	GC33-6683
Programming Guide	SC33-6684
Programming Reference	SC33-6685
Run-Time Migration Guide	SC33-6687
Writing Interlanguage Communication Applications	SC33-6686

VSE/ICCF

Book title	Order number
Administration and Operations	SC33-6738
User's Guide	SC33-6739

VSE/POWER

Book title	Order number
Administration and Operation	SC33-6733
Application Programming	SC33-6736
Networking Guide	SC33-6735
Remote Job Entry User's Guide	SC33-6734

VSE/VSAM

Book title	Order number
Commands	SC33-6731
User's Guide and Application Programming	SC33-6732

VTAM for VSE/ESA

Book title	Order number
Customization	LY43-0063
Diagnosis	LY43-0065
Data Areas	LY43-0104
Messages and Codes	SC31-6493
Migration Guide	GC31-8072
Network Implementation Guide	SC31-6494
Operation	SC31-6495
Overview	GC31-8114
Programming	SC31-6496
Programming for LU6.2	SC31-6497
Release Guide	GC31-8090
Resource Definition Reference	SC31-6498

Books from VSE/ESA 2.4 optional program libraries

C for VSE/ESA (C/VSE)

Book title	Order number
C Run-Time Library Reference	SC33-6689
C Run-Time Programming Guide	SC33-6688
Diagnosis Guide	GC09-2426
Installation and Customization Guide	GC09-2422
Language Reference	SC09-2425
Licensed Program Specification	GC09-2421
Migration Guide	SC09-2423
User's Guide	SC09-2424

COBOL for VSE/ESA (COBOL/VSE)

Book title	Order number
Debug Tool for VSE/ESA Fact Sheet	GC26-8925
Debug Tool for VSE/ESA Installation and Customization Guide	SC26-8798
Debug Tool for VSE/ESA User's Guide and Reference	SC26-8797
Diagnosis Guide	SC26-8528
General Information	GC26-8068
Installation and Customization Guide	SC26-8071
Language Reference	SC26-8073
Licensed Program Specifications	GC26-8069
Migration Guide	GC26-8070
Migrating VSE Applications To Advanced COBOL	GC26-8349
Programming Guide	SC26-8072

DB2 Server for VSE

Book title	Order number
Application Programming	SC09-2393
Database Administration	GC09-2389
Installation	GC09-2391
Interactive SQL Guide and Reference	SC09-2410
Operation	SC09-2401
Overview	GC08-2386
System Administration	GC09-2406

DL/I VSE

Book title	Order number
Application and Database Design	SH24-5022
Application Programming: CALL and RQDLI Interface	SH12-5411
Application Programming: High-Level Programming Interface	SH24-5009
Database Administration	SH24-5011
Diagnostic Guide	SH24-5002
General Information	GH20-1246
Guide for New Users	SH24-5001
Interactive Resource Definition and Utilities	SH24-5029
Library Guide and Master Index	GH24-5008
Licensed Program Specifications	GH24-5031
Low-level Code and Continuity Check Feature	SH20-9046
Library Guide and Master Index	GH24-5008
Messages and Codes	SH12-5414
Recovery and Restart Guide	SH24-5030
Reference Summary: CALL Program Interface	SX24-5103
Reference Summary: System Programming	SX24-5104
Reference Summary: HLPI Interface	SX24-5120
Release Guide	SC33-6211

PL/I for VSE/ESA (PL/I VSE)

Book title	Order number
Compile Time Messages and Codes	SC26-8059
Debug Tool For VSE/ESA User's Guide and Reference	SC26-8797
Diagnosis Guide	SC26-8058
Installation and Customization Guide	SC26-8057
Language Reference	SC26-8054
Licensed Program Specifications	GC26-8055
Migration Guide	SC26-8056
Programming Guide	SC26-8053
Reference Summary	SX26-3836

Screen Definition Facility II (SDF II)

Book title	Order number
VSE Administrator's Guide	SH12-6311
VSE General Introduction	SH12-6315
VSE Primer for CICS/BMS Programs	SH12-6313
VSE Run-Time Services	SH12-6312

Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

The following paragraph does not apply in the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore this statement may not apply to you.

This publication could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact IBM United Kingdom Laboratories, MP151, Hursley Park, Winchester, Hampshire, England, SO21 2JN. Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Programming License Agreement, or any equivalent agreement between us.

Trademarks and service marks

The following terms, used in this publication, are trademarks or service marks of IBM Corporation in the United States or other countries:

CICS	IBM	System/370
CICS/ESA	IMS	VSE/ESA
CICS/VSE	OS/390	VTAM
DB2		

Other company, product, and service names may be the trademarks or service marks of others.

Index

Numerics

- 16MB line 123
- 2260 Display Station
 - CONVERSE command 54
 - RECEIVE command 219
 - SEND command 263
- 2265 Display Station
 - CONVERSE command 54
 - RECEIVE command 219
 - SEND command 263
- 2741 Communication Terminal
 - CONVERSE command 55
 - RECEIVE command 219
 - SEND command 263
- 2770 Data Communication System
 - CONVERSE command 53, 56
 - RECEIVE command 218
 - SEND command 262
- 2780 Data Transmission Terminal
 - CONVERSE command 53, 56
 - RECEIVE command 218
 - SEND command 262
- 2980 General Banking Terminal System
 - DFH2980 structure 221
 - output control 220
 - output to common buffer 221
 - passbook control 220
 - RECEIVE/SEND commands 220, 264
- 3270 Information Display System 222
 - (BTAM supported) 57, 142, 166, 264
 - logical unit 43, 143, 210, 251
- 3600 Finance Communication System
 - 3601 logical unit 43, 211, 252
 - 3614 logical unit 44, 212, 252
 - CONVERSE command 57
 - pipeline logical units 251
 - RECEIVE command 222
 - SEND command 265
- 3630 Plant Communication System
 - RECEIVE command 211
 - SEND command 252
- 3650 Host Command Processor
 - CONVERSE command 46
- 3650 Logical Units
 - RECEIVE command 213
- 3650 Store System
 - host conversational
 - LU 3270 253
 - LU 3653 254
 - interpreter logical unit 45, 150, 155, 212, 253
- 3650/3680 Full-function Logical Unit
 - RECEIVE command 214
- 3650/3680 Full-function Logical Unit (*continued*)
 - SEND command 256
- 3650/3680 Store System
 - host command processor LU 254
- 3660 Supermarket Scanning System
 - CONVERSE command 53
 - RECEIVE command 218
 - SEND command 262
- 3680 Host Command Processor
 - CONVERSE command 46
- 3680 Programmable Store System
 - host command processor LU 254
- 3735 Programmable Buffered Terminal 59, 223, 265
- 3740 Data Entry System 59, 148, 149, 223
- 3767 Communication Terminal
 - interactive logical unit 47, 213, 255
- 3770 Data Communication System
 - batch logical unit 47, 214, 255
- 3770 Full Function Logical Unit
 - RECEIVE command 214
 - SEND command 256
- 3770 interactive logical unit
 - RECEIVE command 213
 - SEND command 255
- 3780 Data Communication Terminal
 - CONVERSE command 53
 - RECEIVE command 218
 - SEND command 262
- 3790 Communication System
 - 3270-display logical unit 48, 224, 257
 - Full Function Logical Unit 48
 - full-function logical unit 214, 256
 - SCS printer logical unit 256
- 4700 Finance Communication System
 - CONVERSE command 57
 - RECEIVE command 222
 - SEND command 265

A

- ABCODE option
 - ABEND command 10
 - ASSIGN command 21
- ABDUMP option
 - ASSIGN command 21
- ABEND AICA timeout 6
- ABEND command 10
- ABEND exit, reactivating 128
- abend support commands 6
- abnormal termination, task 128
- ABPROGRAM option
 - ASSIGN command 21

- absolute expressions 4
- ABSTIME option
 - ASKTIME command 19
 - FORMATTIME command 95
- access to system information
 - ADDRESS command 11
 - ADDRESS SET command 13
 - ASSIGN command 20
 - CICS storage areas 11, 13
- ACCUM option
 - SEND CONTROL command 269
 - SEND MAP command 274
 - SEND TEXT command 282
- ACEE option
 - ADDRESS command 11
- ACTION option
 - WRITE OPERATOR command 365
- ACTPARTN option
 - SEND CONTROL command 270
 - SEND MAP command 274
 - SEND TEXT command 283
- ADDRESS command 11
- ADDRESS SET command 13
- address, cursor 391
- AFTER option
 - POST command 186
 - ROUTE command 244
 - START command 333
- ALARM option
 - SEND CONTROL command 270
 - SEND MAP command 274
 - SEND TEXT command 283
 - SEND TEXT NOEDIT command 289
- ALARM value
 - DFHMDI 423
 - DFHMSD 429
- ALIGNED attribute
 - PL/I 3
- ALL option
 - SEND PAGE command 278
- ALLOCATE (APPC) command 14
- ALLOCATE (LUTYPE6.1) command 16
- ALLOCATE (MRO) command 18
- ALTER option
 - QUERY SECURITY command 190
- ALTERNATE option
 - CONVERSE (non-VTAM) command 60
 - CONVERSE (VTAM) command 49
 - SEND (non-VTAM) command 266
 - SEND (VTAM) command 258
 - SEND CONTROL command 270
 - SEND MAP command 274
 - SEND TEXT command 283
 - SEND TEXT NOEDIT command 289
- ALTSCRN operand
 - DFHPSD 434
- ALTSCRNHT option
 - ASSIGN command 21
- ALTSCRNWD option
 - ASSIGN command 21
- ANYKEY option
 - HANDLE AID command 130
- APLKYBD option
 - ASSIGN command 21
- APLTEXT option
 - ASSIGN command 21
- APPC basic conversations
 - commands 6
- APPC logical unit
 - acquiring session to 14
 - initiating conversation with 38
 - returning mapped sessions to CICS 99
 - sending and receiving 40
- APPC mapped conversations
 - abending 136
 - commands 6
 - ensuring transmission of accumulated data 349
 - extracting attributes of 89
 - informing partner of error 154
 - issuing a positive response 141
 - receiving data 208
 - requesting change of direction 169
 - retrieving values from attach header 92
 - returning sessions to CICS 99
 - sending data 248
- application performance, monitoring 181
- APPLID option
 - ASSIGN command 21
- argument values
 - assembler language 4
 - C 2
 - COBOL 2
 - PL/I 3
- ASA option
 - SPOOLOPEN OUTPUT 307
 - SPOOLOPEN REPORT command 310
- ASIS option
 - CONVERSE (non-VTAM) command 60
 - CONVERSE (VTAM) command 49
 - RECEIVE (non-VTAM) command 225
 - RECEIVE (VTAM) command 215
 - RECEIVE MAP command 228
 - RECEIVE PARTN command 230
 - SEND (non-VTAM) command 266
- ASKIP value
 - DFHMDF 415
- ASKTIME command 19
- ASRAINTRPT option
 - ASSIGN command 21
- ASRAKEY option
 - ASSIGN command 21

- ASRAPSW option
 - ASSIGN command 21
- ASRAREGS option
 - ASSIGN command 21
- ASRASTG option
 - ASSIGN command 22
- assembler language
 - argument values 4
 - LENGTH option default 4
 - program exit 384
 - register contents 384
 - translated code 384
- ASSIGN command 20
- asynchronous interrupt 388
- AT option
 - POST command 186
 - ROUTE command 244
 - START command 333
- attach
 - start a task 336
- ATTACHID option
 - BUILD ATTACH (LUTYPE6.1) command 29
 - BUILD ATTACH (MRO) command 31
 - CONVERSE (non-VTAM) command 60
 - CONVERSE (VTAM) command 49
 - EXTRACT ATTACH (LUTYPE6.1) command 85
 - EXTRACT ATTACH (MRO) command 87
 - SEND (non-VTAM) command 266
 - SEND (VTAM) command 258
- attention identifier (AID) 130
- ATTRB operand
 - DFHMDf 415
 - DFHPDI 433
- attributes
 - control character list, DFHBMSCA 405
- authentication commands 6
- AUTOPAGE option
 - SEND PAGE command 278
- AUXILIARY option
 - WRITEQ TS command 369

B

- back out to a syncpoint 342
- BASE operand
 - DFHMSD 428
- BASE value
 - DFHMDf 419
 - DFHMDI 425
 - DFHMSD 431
- basic mapping support (BMS)
 - commands 7
 - completing a logical message 278
 - deleting a logical message 188
 - determining input partition 230
 - field definition macro 411, 415

- basic mapping support (BMS) (*continued*)
 - full BMS
 - RECEIVE MAP command 228
 - RECEIVE PARTN command 230
 - SEND CONTROL command 269
 - SEND MAP command 273
 - SEND PAGE 278
 - SEND PARTNSET 281
 - SEND TEXT command 282
 - SEND TEXT MAPPED 286
 - SEND TEXT NOEDIT 288
 - full-function BMS
 - PURGE MESSAGE 188
 - map definition macro 411, 422
 - mapping input data 228
 - mapset definition macro 411, 428
 - minimum BMS
 - RECEIVE MAP command 228
 - SEND CONTROL command 269
 - SEND MAP command 273
 - partition definition macro 412, 433
 - partition set definition macro 412, 434
 - related constants 405
 - routing a logical message 244
 - sending previously mapped data 286
 - sending user-defined data stream 288
 - standard BMS
 - RECEIVE MAP command 228
 - RECEIVE PARTN command 230
 - SEND CONTROL command 269
 - SEND MAP command 273
 - SEND PARTNSET 281
 - SEND TEXT command 282
- batch data interchange (BDI)
 - add record to data set 139
 - commands 6
 - conditions 156
 - delete a record from data set 151
 - read record from data set 162
 - request next record number 156
 - send data to output device 167
 - terminate data set 137, 146
 - update a record in data set 164
 - wait for function completion 171
- batch logical unit, 3770 47, 214, 255
- batch mode application, 3740 59, 223, 266
- BELOW option
 - GETMAIN command 123
- BIF DEEDIT command 28
- BLANK value
 - DFHMDf 417
- BLINK value
 - DFHMDf 417
 - DFHMDI 424
 - DFHMSD 430

BLOCK value
 DFHMDI 423
 DFHMSD 429

BMS
 See basic mapping support

BOTHNUM option
 SPOOLOPEN REPORT command 310

BOTTOM value
 DFHMDI 424

browse operation
 ending 78
 read next record 197
 read previous record 201
 reset starting point 234
 starting 337

BRT value
 DFHMDF 415

BTAM programmable terminals 388

BTRANS option
 ASSIGN command 22

BUFFER option
 GDS RECEIVE command 117
 RECEIVE (non-VTAM) command 225
 RECEIVE (VTAM) command 215

BUFSIZE operand
 DFHPDI 433

BUILD ATTACH (LUTYPE6.1) command 29

BUILD ATTACH (MRO) command 31

built-in functions
 commands 7

C

C language
 ADDRESS COMMAREA 11
 ADDRESS EIB 11
 argument values 2
 LENGTH option default 3
 translated code 383

CANCEL command 33

CANCEL option
 ABEND command 10
 HANDLE ABEND command 128

CARD option
 ISSUE ABORT command 137
 ISSUE END command 146
 ISSUE SEND command 167
 ISSUE WAIT command 171

CASE operand
 DFHMDF 416

CBIDERR condition
 ALLOCATE (APPC) command 14
 ALLOCATE (LUTYPE6.1) command 16
 CONVERSE (non-VTAM) command 61
 CONVERSE (VTAM) command 50
 EXTRACT ATTACH (LUTYPE6.1) command 86

CBIDERR condition (*continued*)
 EXTRACT ATTACH (MRO) command 88
 SEND (non-VTAM) command 267
 SEND (VTAM) command 259

CBUFF option
 SEND (non-VTAM) command 266

CCERROR condition
 SPOOLOPEN ESCAPE command 300
 SPOOLOPEN MAPNAME command 305
 SPOOLOPEN REPORT (JCL) command 315
 SPOOLOPEN REPORT (SCS/T3270) command 320
 SPOOLOPEN REPORT command 314
 SPOOLWRITE MAPNAME command 327
 SPOOLWRITE REPORT command 330

CHANGE PASSWORD command 35

CHANGE TASK command 36

CHANGETIME option
 VERIFY PASSWORD command 347

CHARSZ operand
 DFHPDI 433
 DFHPSD 434

CHECKPOINT, full form of CHKP (DL/I) command 37

CHKP (DL/I) command 37

CICSDATAKEY option
 GETMAIN command 123

CLASS option
 SPOOLOPEN ESCAPE command 298
 SPOOLOPEN INPUT 301
 SPOOLOPEN MAPNAME command 303
 SPOOLOPEN OUTPUT 307
 SPOOLOPEN REPORT (SCS/T3270) command 317
 SPOOLOPEN REPORT command 310

CLEAR option
 HANDLE AID command 130

CLRPARTN option
 HANDLE AID command 130

CMDSEC option
 ASSIGN command 22

CNOTCOMPL option
 SEND (non-VTAM) command 266
 SEND (VTAM) command 258

COBOL
 argument values 2
 translated code 383

CODEREG operand 385

COLOR operand
 DFHMDF 416
 DFHMDI 422
 DFHMSD 428

COLOR option
 ASSIGN command 22

COLUMN operand
 DFHMDI 422

column value
 DFHMDI 425

command language translator
 translated code 384
 commands
 format, arguments 1
 printer spooling 309
 report controller 309
 security 8
 spool 8
 temporary storage control 8
 COMMAREA option
 ADDRESS command 11
 LINK command 175
 RETURN command 240
 XCTL command 371
 common buffer, output to, 2980 221
 common programming interface communications (CPI
 communications) 395
 COMPLETE option
 DUMP TRANSACTION command 75
 CONFIRM option
 GDS SEND command 119
 SEND (VTAM) command 258
 CONNECT PROCESS command 38
 CONSOLE option
 ISSUE ABORT command 137
 ISSUE END command 146
 ISSUE SEND command 167
 ISSUE WAIT command 171
 console support commands 7
 constants
 AID values, DFHAID 409
 attribute values, DFHBMSCA 405
 for 3270 attributes 405
 for examining EIBAID field 409
 for MSR control values 407
 for printer format controls 405
 MSR control, DFHtex read 407
 printer control values, DFHBMSCA 405
 CONTROL option
 QUERY SECURITY command 190
 CONVDATA option
 GDS CONNECT PROCESS command 107
 GDS EXTRACT ATTRIBUTES command 109
 GDS FREE command 111
 GDS ISSUE ABEND command 112
 GDS ISSUE CONFIRMATION command 113
 GDS ISSUE ERROR command 114
 GDS ISSUE PREPARE command 115
 GDS ISSUE SIGNAL command 116
 GDS RECEIVE command 117
 GDS SEND command 119
 GDS WAIT command 121
 CONVERSE (2260) command 54
 CONVERSE (2741) command 55
 CONVERSE (2770) command 56
 CONVERSE (2780) command 56
 CONVERSE (3270 display) command 57
 CONVERSE (3270 logical) command 43
 CONVERSE (3600 BTAM) command 57
 CONVERSE (3600-3601) command 43
 CONVERSE (3600-3614) command 44
 CONVERSE (3650 interpreter) command 45
 CONVERSE (3650-3270) command 45
 CONVERSE (3650-3653) command 46
 CONVERSE (3650-3680) command 46
 CONVERSE (3735) command 59
 CONVERSE (3740) command 59
 CONVERSE (3767) command 47
 CONVERSE (3770) command 47
 CONVERSE (3790 3270-display) command 48
 CONVERSE (3790 full-function or inquiry) command 48
 CONVERSE (APPC) command 40
 CONVERSE (LUTYPE2/LUTYPE3) command 41
 CONVERSE (LUTYPE4) command 41
 CONVERSE (LUTYPE6.1) command 42
 CONVERSE (MRO) command 52
 CONVERSE (non-VTAM default) command 52
 CONVERSE (SCS) command 42
 CONVERSE (System/3) command 53
 CONVERSE (System/7) command 53
 CONVERSE (VTAM default) command 40
 CONVERSE option
 ISSUE LOAD command 155
 converse with terminal or LU 388
 CONVID option
 CONNECT PROCESS command 38
 CONVERSE (non-VTAM) command 60
 CONVERSE (VTAM) command 49
 EXTRACT ATTACH (LUTYPE6.1) command 85
 EXTRACT ATTACH (MRO) command 87
 EXTRACT ATTRIBUTES (APPC) command 89
 EXTRACT ATTRIBUTES (MRO) command 90
 EXTRACT PROCESS command 92
 FREE (APPC) command 99
 FREE (LUTYPE6.1) command 100
 FREE (MRO) command 101
 GDS ALLOCATE command 104
 GDS CONNECT PROCESS command 107
 GDS EXTRACT ATTRIBUTES command 109
 GDS EXTRACT PROCESS command 110
 GDS FREE command 111
 GDS ISSUE ABEND command 112
 GDS ISSUE CONFIRMATION command 113
 GDS ISSUE ERROR command 114
 GDS ISSUE PREPARE command 115
 GDS ISSUE SIGNAL command 116
 GDS RECEIVE command 117
 GDS SEND command 119
 GDS WAIT command 121
 ISSUE ABEND command 136
 ISSUE CONFIRMATION command 141

CONVID option (*continued*)
 ISSUE ERROR command 154
 ISSUE PREPARE command 159
 ISSUE SIGNAL (APPC) command 169
 ISSUE SIGNAL (LUTYPE6.1) command 170
 POINT command 183
 RECEIVE (VTAM) command 215
 SEND (VTAM) command 258
 WAIT CONVID command 349
 WAIT TERMINAL command 357

COPIES option
 SPOOLOPEN ESCAPE command 298
 SPOOLOPEN MAPNAME command 303
 SPOOLOPEN REPORT (SCS/T3270) command 317
 SPOOLOPEN REPORT command 311

copy displayed information 390

copybooks
 DFHAID 409
 DFHBMSCA 405
 DFHEIBLK 384
 DFHMSRCA 407

CPI communications (SAA) 395

create a journal record 173, 363

CSMT transaction
 resend message destination 58

CSTL transaction
 resend message destination 58

CTLCHAR option
 CONVERSE (non-VTAM) command 60
 CONVERSE (VTAM) command 49
 ISSUE COPY (3270 display) command 142
 ISSUE COPY (3270 logical) command 143
 SEND (non-VTAM) command 266
 SEND (VTAM) command 258

CTRL operand
 DFHMDI 422
 DFHMSD 428

CURRENT option
 SEND PAGE command 278

CURSLOC operand
 DFHMDI 423
 DFHMSD 429

cursor address 391

CURSOR option
 SEND CONTROL command 270
 SEND MAP command 274
 SEND TEXT command 283

cursor position
 terminal control 391

CVDA (CICS-value data area)
 argument values 2
 command format 2
 passing and receiving 5
 values in alphabetic order 401
 values in numeric order 401

CVDA options

ACTION
 WRITE OPERATOR command 365

ALTER
 QUERY SECURITY command 190

ASRAKEY
 ASSIGN command 21

CONTROL
 QUERY SECURITY command 190

LOGMESSAGE
 QUERY SECURITY command 190

MAXLIFETIME
 DEQ 71
 ENQ 80

PURGEABILITY
 WAIT EXTERNAL 352
 WAITCICS 358

READ
 QUERY SECURITY command 190

STATE 50, 61, 216, 226, 259, 267
 ALLOCATE (APPC) 14
 ALLOCATE (MRO) 18
 CONNECT PROCESS 38
 EXTRACT ATTRIBUTES (APPC) 89
 EXTRACT ATTRIBUTES (MRO) 90
 FREE (APPC) 99
 FREE (MRO) 101
 GDS ALLOCATE 104
 GDS CONNECT PROCESS 107
 GDS EXTRACT ATTRIBUTES 109
 GDS FREE 111
 GDS ISSUE ABEND 112
 GDS ISSUE CONFIRMATION 113
 GDS ISSUE ERROR 114
 GDS ISSUE PREPARE 115
 GDS ISSUE SIGNAL 116
 GDS RECEIVE 118
 GDS SEND 119
 GDS WAIT 121
 ISSUE ABEND command 136
 ISSUE CONFIRMATION 141
 ISSUE ERROR 154
 ISSUE PREPARE 159
 ISSUE SIGNAL (APPC) 169
 WAIT CONVID 349

UPDATE
 QUERY SECURITY command 191

CVDA values

ALLOCATED
 ALLOCATE (APPC) 14
 ALLOCATE (MRO) 18
 CONNECT PROCESS 38
 EXTRACT ATTRIBUTES (APPC) 89
 EXTRACT ATTRIBUTES (MRO) 90
 FREE (APPC) 99
 FREE (MRO) 101
 GDS ALLOCATE 104

CVDA values (continued)

ALLOCATED (continued)

GDS CONNECT PROCESS 107
 GDS EXTRACT ATTRIBUTES 109
 GDS FREE 111
 GDS ISSUE ABEND 112
 GDS ISSUE CONFIRMATION 113
 GDS ISSUE ERROR 114
 GDS ISSUE PREPARE 115
 GDS ISSUE SIGNAL 116
 GDS RECEIVE 118
 GDS SEND 119
 GDS WAIT 121
 ISSUE ABEND command 136
 ISSUE CONFIRMATION 141
 ISSUE ERROR 154
 ISSUE PREPARE 159
 ISSUE SIGNAL (APPC) 169
 RECEIVE (APPC) command 216
 RECEIVE (MRO) command 226
 SEND (APPC) command 259
 SEND (MRO) command 267
 WAIT CONVID 349

ALTERABLE

QUERY SECURITY command 190

CICSEXECKEY

ASSIGN command 21

CONFFREE

CONNECT PROCESS 38
 EXTRACT ATTRIBUTES (APPC) 89
 FREE (APPC) 99
 GDS ALLOCATE 104
 GDS CONNECT PROCESS 107
 GDS EXTRACT ATTRIBUTES 109
 GDS FREE 111
 GDS ISSUE ABEND 112
 GDS ISSUE CONFIRMATION 113
 GDS ISSUE ERROR 114
 GDS ISSUE PREPARE 115
 GDS ISSUE SIGNAL 116
 GDS RECEIVE 118
 GDS SEND 119
 GDS WAIT 121
 ISSUE ABEND command 136
 ISSUE CONFIRMATION 141
 ISSUE ERROR 154
 ISSUE PREPARE 159
 ISSUE SIGNAL (APPC) 169
 RECEIVE (APPC) command 216
 SEND (APPC) command 259
 WAIT CONVID 349

CONFRECEIVE

CONNECT PROCESS 38
 EXTRACT ATTRIBUTES (APPC) 89
 FREE (APPC) 99
 GDS ALLOCATE 104
 GDS CONNECT PROCESS 107

CVDA values (continued)

CONFRECEIVE (continued)

GDS EXTRACT ATTRIBUTES 109
 GDS FREE 111
 GDS ISSUE ABEND 112
 GDS ISSUE CONFIRMATION 113
 GDS ISSUE ERROR 114
 GDS ISSUE PREPARE 115
 GDS ISSUE SIGNAL 116
 GDS RECEIVE 118
 GDS SEND 119
 GDS WAIT 121
 ISSUE ABEND command 136
 ISSUE CONFIRMATION 141
 ISSUE ERROR 154
 ISSUE PREPARE 159
 ISSUE SIGNAL (APPC) 169
 RECEIVE (APPC) command 216
 SEND (APPC) command 259
 WAIT CONVID 349

CONFSEND

CONNECT PROCESS 38
 EXTRACT ATTRIBUTES (APPC) 89
 FREE (APPC) 99
 GDS ALLOCATE 104
 GDS CONNECT PROCESS 107
 GDS EXTRACT ATTRIBUTES 109
 GDS FREE 111
 GDS ISSUE ABEND 112
 GDS ISSUE CONFIRMATION 113
 GDS ISSUE ERROR 114
 GDS ISSUE PREPARE 115
 GDS ISSUE SIGNAL 116
 GDS RECEIVE 118
 GDS SEND 119
 GDS WAIT 121
 ISSUE ABEND command 136
 ISSUE CONFIRMATION 141
 ISSUE ERROR 154
 ISSUE PREPARE 159
 ISSUE SIGNAL (APPC) 169
 RECEIVE (APPC) command 216
 SEND (APPC) command 259
 WAIT CONVID 349

CRITICAL

WRITE OPERATOR command 365

CTRLABLE

QUERY SECURITY command 190

EVENTUAL

WRITE OPERATOR command 365

FREE

CONNECT PROCESS 38
 EXTRACT ATTRIBUTES (APPC) 89
 EXTRACT ATTRIBUTES (MRO) 90
 FREE (APPC) 99
 FREE (MRO) 101
 GDS ALLOCATE 104

CVDA values (continued)

FREE (continued)
 GDS CONNECT PROCESS 107
 GDS EXTRACT ATTRIBUTES 109
 GDS FREE 111
 GDS ISSUE ABEND 112
 GDS ISSUE CONFIRMATION 113
 GDS ISSUE ERROR 114
 GDS ISSUE PREPARE 115
 GDS ISSUE SIGNAL 116
 GDS RECEIVE 118
 GDS SEND 119
 GDS WAIT 121
 ISSUE ABEND command 136
 ISSUE CONFIRMATION 141
 ISSUE ERROR 154
 ISSUE PREPARE 159
 ISSUE SIGNAL (APPC) 169
 RECEIVE (APPC) command 216
 RECEIVE (MRO) command 226
 SEND (APPC) command 259
 SEND (MRO) command 267
 WAIT CONVID 349
 IMMEDIATE
 WRITE OPERATOR command 365
 LOG
 QUERY SECURITY command 190
 LUW
 DEQ 71
 ENQ 80
 NOLOG
 QUERY SECURITY command 190
 NONCICS
 ASSIGN command 21
 NOTALTERABLE
 QUERY SECURITY command 190
 NOTAPPLIC
 ASSIGN command 21
 NOTCTRLABLE
 QUERY SECURITY command 190
 NOTPURGEABLE
 WAIT EXTERNAL 352
 WAITCICS 358
 NOTREADABLE
 QUERY SECURITY command 190
 NOTUPDATABLE
 QUERY SECURITY command 191
 PENDFREE
 CONNECT PROCESS 38
 EXTRACT ATTRIBUTES (APPC) 89
 EXTRACT ATTRIBUTES (MRO) 90
 FREE (APPC) 99
 FREE (MRO) 101
 GDS ALLOCATE 104
 GDS CONNECT PROCESS 107
 GDS EXTRACT ATTRIBUTES 109
 GDS FREE 111

CVDA values (continued)

PENDFREE (continued)
 GDS ISSUE ABEND 112
 GDS ISSUE CONFIRMATION 113
 GDS ISSUE ERROR 114
 GDS ISSUE PREPARE 115
 GDS ISSUE SIGNAL 116
 GDS RECEIVE 118
 GDS SEND 119
 GDS WAIT 121
 ISSUE ABEND command 136
 ISSUE CONFIRMATION 141
 ISSUE ERROR 154
 ISSUE PREPARE 159
 ISSUE SIGNAL (APPC) 169
 RECEIVE (APPC) command 216
 RECEIVE (MRO) command 226
 SEND (APPC) command 259
 SEND (MRO) command 267
 WAIT CONVID 349
 PENDRECEIVE
 CONNECT PROCESS 38
 EXTRACT ATTRIBUTES (APPC) 89
 FREE (APPC) 99
 GDS ALLOCATE 104
 GDS CONNECT PROCESS 107
 GDS EXTRACT ATTRIBUTES 109
 GDS FREE 111
 GDS ISSUE ABEND 112
 GDS ISSUE CONFIRMATION 113
 GDS ISSUE ERROR 114
 GDS ISSUE PREPARE 115
 GDS ISSUE SIGNAL 116
 GDS RECEIVE 118
 GDS SEND 119
 GDS WAIT 121
 ISSUE ABEND command 136
 ISSUE CONFIRMATION 141
 ISSUE ERROR 154
 ISSUE PREPARE 159
 ISSUE SIGNAL (APPC) 169
 RECEIVE (APPC) command 216
 SEND (APPC) command 259
 WAIT CONVID 349
 PURGEABLE
 WAIT EXTERNAL 352
 WAITCICS 358
 READABLE
 QUERY SECURITY command 190
 RECEIVE
 CONNECT PROCESS 38
 CONVERSE command (MRO) 61
 EXTRACT ATTRIBUTES (APPC) 89
 EXTRACT ATTRIBUTES (MRO) 90
 FREE (APPC) 99
 FREE (MRO) 101
 GDS ALLOCATE 104

CVDA values (continued)

RECEIVE (continued)

GDS CONNECT PROCESS 107
GDS EXTRACT ATTRIBUTES 109
GDS FREE 111
GDS ISSUE ABEND 112
GDS ISSUE CONFIRMATION 113
GDS ISSUE ERROR 114
GDS ISSUE PREPARE 115
GDS ISSUE SIGNAL 116
GDS RECEIVE 118
GDS SEND 120
GDS WAIT 121
ISSUE ABEND command 136
ISSUE CONFIRMATION 141
ISSUE ERROR 154
ISSUE PREPARE 159
ISSUE SIGNAL (APPC) 169
RECEIVE (APPC) command 216
RECEIVE (MRO) command 226
SEND (APPC) command 259
SEND (MRO) command 267
WAIT CONVID 349

ROLLBACK

CONNECT PROCESS 38
CONVERSE command (MRO) 61
EXTRACT ATTRIBUTES (APPC) 89
EXTRACT ATTRIBUTES (MRO) 90
FREE (APPC) 99
FREE (MRO) 101
GDS ALLOCATE 104
GDS CONNECT PROCESS 107
GDS EXTRACT ATTRIBUTES 109
GDS FREE 111
GDS ISSUE ABEND 112
GDS ISSUE CONFIRMATION 113
GDS ISSUE ERROR 114
GDS ISSUE PREPARE 115
GDS ISSUE SIGNAL 116
GDS RECEIVE 118
GDS SEND 120
GDS WAIT 121
ISSUE ABEND command 136
ISSUE CONFIRMATION 141
ISSUE ERROR 154
ISSUE PREPARE 159
ISSUE SIGNAL (APPC) 169
RECEIVE (APPC) command 216
RECEIVE (MRO) command 226
SEND (APPC) command 259
SEND (MRO) command 267
WAIT CONVID 349

SEND

CONNECT PROCESS 38
CONVERSE command (MRO) 61
EXTRACT ATTRIBUTES (APPC) 89
EXTRACT ATTRIBUTES (MRO) 90

CVDA values (continued)

SEND (continued)

FREE (APPC) 99
FREE (MRO) 101
GDS ALLOCATE 104
GDS CONNECT PROCESS 107
GDS EXTRACT ATTRIBUTES 109
GDS FREE 111
GDS ISSUE ABEND 112
GDS ISSUE CONFIRMATION 113
GDS ISSUE ERROR 114
GDS ISSUE PREPARE 115
GDS ISSUE SIGNAL 116
GDS RECEIVE 118
GDS SEND 120
GDS WAIT 121
ISSUE ABEND command 136
ISSUE CONFIRMATION 141
ISSUE ERROR 154
ISSUE PREPARE 159
ISSUE SIGNAL (APPC) 169
RECEIVE (APPC) command 216
RECEIVE (MRO) command 226
SEND (APPC) command 259
SEND (MRO) command 267
WAIT CONVID 349

SYNCFREE

CONNECT PROCESS 38
CONVERSE command (MRO) 61
EXTRACT ATTRIBUTES (APPC) 89
EXTRACT ATTRIBUTES (MRO) 90
FREE (APPC) 99
FREE (MRO) 101
GDS ALLOCATE 104
GDS CONNECT PROCESS 107
GDS EXTRACT ATTRIBUTES 109
GDS FREE 111
GDS ISSUE ABEND 112
GDS ISSUE CONFIRMATION 113
GDS ISSUE ERROR 114
GDS ISSUE PREPARE 115
GDS ISSUE SIGNAL 116
GDS RECEIVE 118
GDS SEND 120
GDS WAIT 121
ISSUE ABEND command 136
ISSUE CONFIRMATION 141
ISSUE ERROR 154
ISSUE PREPARE 159
ISSUE SIGNAL (APPC) 169
RECEIVE (APPC) command 216
RECEIVE (MRO) command 226
SEND (APPC) command 259
SEND (MRO) command 267
WAIT CONVID 349

SYNCRECEIVE

CONNECT PROCESS 38

CVDA values (continued)

SYNCRECEIVE (continued)

CONVERSE command (MRO) 61
EXTRACT ATTRIBUTES (APPC) 89
EXTRACT ATTRIBUTES (MRO) 90
FREE (APPC) 99
FREE (MRO) 101
GDS ALLOCATE 104
GDS CONNECT PROCESS 107
GDS EXTRACT ATTRIBUTES 109
GDS FREE 111
GDS ISSUE ABEND 112
GDS ISSUE CONFIRMATION 113
GDS ISSUE ERROR 114
GDS ISSUE PREPARE 115
GDS ISSUE SIGNAL 116
GDS RECEIVE 118
GDS SEND 120
GDS WAIT 121
ISSUE ABEND command 136
ISSUE CONFIRMATION 141
ISSUE ERROR 154
ISSUE PREPARE 159
ISSUE SIGNAL (APPC) 169
RECEIVE (APPC) command 216
RECEIVE (MRO) command 226
SEND (APPC) command 259
SEND (MRO) command 267
WAIT CONVID 349

SYNCSEND

CONNECT PROCESS 38
CONVERSE command (MRO) 61
EXTRACT ATTRIBUTES (APPC) 89
EXTRACT ATTRIBUTES (MRO) 90
FREE (APPC) 99
FREE (MRO) 101
GDS ALLOCATE 104
GDS CONNECT PROCESS 107
GDS EXTRACT ATTRIBUTES 109
GDS FREE 111
GDS ISSUE ABEND 112
GDS ISSUE CONFIRMATION 113
GDS ISSUE ERROR 114
GDS ISSUE PREPARE 115
GDS ISSUE SIGNAL 116
GDS RECEIVE 118
GDS SEND 120
GDS WAIT 121
ISSUE ABEND command 136
ISSUE CONFIRMATION 141
ISSUE ERROR 154
ISSUE PREPARE 159
ISSUE SIGNAL (APPC) 169
RECEIVE (APPC) command 216
RECEIVE (MRO) command 226
SEND (APPC) command 259
SEND (MRO) command 267

CVDA values (continued)

SYNCSEND (continued)

WAIT CONVID 349
TASK
DEQ 71
ENQ 81
UPDATABLE
QUERY SECURITY command 191
USEREXECKEY
ASSIGN command 21
CWA option
ADDRESS command 11
CWALENG option
ASSIGN command 22

D

data

passing to new tasks 333

DATA operand

DFHMDI 423
DFHMSD 429

DATA option

FREEMAIN command 102

data sets

add records to 139
interrogating 161
processing termination 146
read records from 162
update records 164

data tables

CICS/user-maintained

ENDBR command 78
READ command 192
READNEXT command 197
READPREV command 201
RESETBR command 234
REWRITE command 242
STARTBR command 337
UNLOCK command 345
WRITE command 360

data to output device, sending 167

data-area argument

CICS command format 2
definition 1

data-value argument

CICS command format 2
definition 1

data, deleting

file control records 65
temporary storage queues 70
transient data queues 69

DATA1 option

MONITOR command 181

DATA2 option

MONITOR command 181

DATALENGTH option
 LINK command 175
 DATAONLY option
 SEND MAP command 274
 DATAPOINTER option
 FREEMAIN command 102
 DATAREG operand 385
 DATASET, valid but FILE preferred 6
 DATASTR option
 BUILD ATTACH (LUTYPE6.1) command 29
 BUILD ATTACH (MRO) command 31
 EXTRACT ATTACH (LUTYPE6.1) command 85
 EXTRACT ATTACH (MRO) command 87
 DATE option
 FORMATTIME command 95
 DATEFORM option
 FORMATTIME command 95
 DATESEP option
 FORMATTIME command 95
 DATETIME option
 SPOOLOPEN REPORT command 311
 DAYCOUNT option
 FORMATTIME command 96
 DAYOFMONTH option
 FORMATTIME command 96
 DAYOFWEEK option
 FORMATTIME command 96
 DAYSLEFT option
 VERIFY PASSWORD command 347
 DCT option
 DUMP TRANSACTION command 75
 DDMMYY option
 FORMATTIME command 96
 DDMMYYYY option
 FORMATTIME command 96
 DEBKEY option
 READ command 192
 STARTBR command 337
 DEBREC option
 READ command 192
 STARTBR command 337
 DEFAULT option
 CONVERSE (non-VTAM) command 60
 CONVERSE (VTAM) command 49
 SEND (non-VTAM) command 266
 SEND (VTAM) command 258
 SEND CONTROL command 270
 SEND MAP command 274
 SEND TEXT command 283
 SEND TEXT NOEDIT command 289
 DEFRESP option
 CONVERSE (non-VTAM) command 60
 CONVERSE (VTAM) command 49
 ISSUE ADD command 139
 ISSUE ERASE command 151
 ISSUE REPLACE command 164
 DEFRESP option (*continued*)
 ISSUE SEND command 167
 SEND (non-VTAM) command 266
 SEND (VTAM) command 258
 DEFSCRNHT option
 ASSIGN command 22
 DEFSCRNWD option
 ASSIGN command 22
 DELAY command 63
 delay processing, task 63
 DELETE command 65
 delete loaded program 232
 DELETE option
 SPOOLCLOSE REPORT command 295
 delete records
 batch data interchange records 151
 DELETE, full form of DLET (DL/I) command 73
 DELETEQ TD command 69
 DELETEQ TS command 70
 deleting data
 temporary storage queues 70
 transient data queues 69
 DELIMITER option
 ASSIGN command 22
 DEQ command 71
 dequeue from resource 71
 DESTCOUNT option
 ASSIGN command 22
 DESTID option
 ASSIGN command 22
 ISSUE ABORT command 137
 ISSUE ADD command 139
 ISSUE END command 146
 ISSUE ERASE command 151
 ISSUE NOTE command 156
 ISSUE QUERY command 161
 ISSUE REPLACE command 164
 ISSUE SEND command 167
 ISSUE WAIT command 171
 DESTIDLENG option
 ASSIGN command 22
 ISSUE ABORT command 137
 ISSUE ADD command 139
 ISSUE END command 146
 ISSUE ERASE command 151
 ISSUE NOTE command 156
 ISSUE QUERY command 161
 ISSUE REPLACE command 164
 ISSUE SEND command 167
 ISSUE WAIT command 171
 DESTINATION option
 SPOOLOPEN ESCAPE command 298
 SPOOLOPEN MAPNAME command 303
 SPOOLOPEN REPORT (SCS/T3270) command 317
 SPOOLOPEN REPORT command 311

- DET value
 - DFHMDF 415
- DFH2980 structure 221
- DFHAID attention identifier list 409
- DFHBMSCA, standard attribute and printer control character list, BMS 405
- DFHEAI interface processor 384
- DFHECALL macro 384
- DFHEIBLK copybook 384
- DFHEICAL macro, use DFHECALL 384
- DFHEIEND macro 384
- DFHEIENT macro
 - CODEREG 385
 - DATAREG 385
 - defaults 385
 - description 384
 - EIBREG 385
- DFHEIGBL macro 384
- DFHEIPLR symbolic register 386
- DFHEIRET macro 384
- DFHEISTG macro 384
- DFHMDF macro 414
- DFHMMDI macro 421
- DFHMIRS 175
- DFHMMSD macro 427
- DFHMMSRCA, MSR control value constants 407
- DFHPDI macro 433
- DFHPSD macro 434
- DFHRESP, built-in function 5
- DFHVALUE, translator routine 5
- diagnostic services commands 7
- DISABLED condition
 - DELETE command 66
 - DELETEQ TD command 69
 - READ command 194
 - READQ TD command 205
 - STARTBR command 338
 - UNLOCK command 345
 - WRITE command 361
 - WRITEQ TD command 367
- disconnect a switched line 388
- DISP option
 - SPOOLCLOSE REPORT command 295
- display-device operations
 - attention identifier (AID) 391
 - attention identifier list, DFHAID 409
 - copy displayed information 390
 - cursor address 391
 - erase all unprotected fields 390
 - input operation without data 390
 - pass control on receipt of an AID 130, 134
 - print displayed information 390
 - standard attribute and printer control character list, DFHBMSCA 405
 - terminal 390

- distributed program link (DPL)
 - ogram link 397
- DL/I services
 - commands 7
- DLET (DL/I) command 73
- DPL, distributed program link 397
- DRK value
 - DFHMDF 416
- DS3270 option
 - ASSIGN command 23
- DSATTS operand
 - DFHMMDI 423
 - DFHMMSD 429
- DSECT value
 - DFHMMSD 432
- DSSCS option
 - ASSIGN command 22
- DSSTAT condition
 - ISSUE RECEIVE command 162
- DUMP command 74
- DUMP TRANSACTION command 75
- DUMPCODE option
 - DUMP TRANSACTION command 75
- DUMPID option
 - DUMP TRANSACTION command 75
- DUPKEY condition
 - DELETE command 66
 - READ command 194
 - READNEXT command 199
 - READPREV command 202
- DUPREC condition
 - REWRITE command 242
 - WRITE command 361
- dynamic storage, extensions 385

E

- ECADDR option
 - WAIT EVENT command 350
- ECBLIST option
 - WAIT EXTERNAL command 352
 - WAITCICS command 358
- EDF, execution diagnostic facility 335
- EIB fields
 - EIBAID 373
 - EIBATT 373
 - EIBCALEN 373
 - EIBCOMPL 373
 - EIBCONF 373
 - EIBCPOSN 373
 - EIBDATE 373
 - EIBDS 374
 - EIBEOC 374
 - EIBERR 374
 - EIBERRCD 374
 - EIBFMH 374

EIB fields (*continued*)

- EIBFN 374
- EIBFREE 375
- EIBNODAT 375
- EIBRCODE 376
- EIBRECV 378
- EIBREQID 379
- EIBRESP 379
- EIBRESP2 379
- EIBRLDBK 379
- EIBRSRCE 379
- EIBSIG 380
- EIBSYNC 380
- EIBSYNRB 380
- EIBTASKN 380
- EIBTIME 380
- EIBTRMID 380
- EIBTRNID 380

EIB option

- ADDRESS command 11

EIBAID 391

- examining contents of field 409

EIBREG operand 385

ENDBR command 78

ENDDATA condition

- RETRIEVE command 238

ENDFILE condition

- READNEXT command 199
- READPREV command 202
- SPOOLREAD command 323

ENDFILE option

- ISSUE ENDOUTPUT command 149

ENDINPT condition

- RECEIVE (non-VTAM) command 226

ENDOUTPUT option

- ISSUE ENDFILE command 148

English and katakana characters, mixed 60, 228, 230

ENQ command 80

ENQBUSY condition

- ENQ command 81

ENTER option

- HANDLE AID command 130

ENTER TRACEID command 82

- monitoring aspects replaced by MONITOR command 181
- tracing aspects replaced by ENTER TRACENUM command 83

ENTER TRACENUM command 83

ENTRY option

- LOAD command 178

entry to assembler-language program 384

ENTRYNAME option

- MONITOR command 181

ENVDEFERR condition

- RETRIEVE command 238

environment services

- commands 7

EOC condition

- ALLOCATE (LUTYPE6.1) command 16
- CONVERSE (non-VTAM) command 61
- CONVERSE (VTAM) command 50
- ISSUE RECEIVE command 162
- RECEIVE (non-VTAM) command 226
- RECEIVE (VTAM) command 216
- RECEIVE MAP command 229
- RECEIVE PARTN command 231

EODS condition

- CONVERSE (VTAM) command 50
- ISSUE RECEIVE command 163
- RECEIVE (VTAM) command 216
- RECEIVE MAP command 229
- RECEIVE PARTN command 231

EOF condition

- CONVERSE (non-VTAM) command 61
- RECEIVE (non-VTAM) command 226

EQUAL option

- READ command 192
- RESETBR command 234
- STARTBR command 337

equated symbols 4

erase all unprotected fields 390

ERASE option

- CONVERSE (non-VTAM) command 60
- CONVERSE (VTAM) command 49
- SEND (non-VTAM) command 266
- SEND (VTAM) command 258
- SEND CONTROL command 270
- SEND MAP command 274
- SEND TEXT command 283
- SEND TEXT NOEDIT command 289

ERASEAUP option

- SEND CONTROL command 270
- SEND MAP command 274

ERRTERM option

- ROUTE command 244

ESCAPE option

- SPOOLOPEN ESCAPE command 298

ESDS (entry-sequenced data set)

- DELETE command 66
- READ 193
- STARTBR command 338
- WRITE command 360

ESM

- ACEE pointer 11
- QUERY SECURITY command, NOTFND condition 191
- QUERY SECURITY command, RESCLASS option 190
- USERNAME 26

ESM, external security manager 335

ESMREASON option

- CHANGE PASSWORD command 35
- SIGNON command 292

ESMREASON option (*continued*)
 VERIFY PASSWORD command 347

ESMRESP option
 CHANGE PASSWORD command 35
 SIGNON command 292
 VERIFY PASSWORD command 347

events, timer
 control area, timer 185
 monitoring point 181
 waiting for 350

EWASUPP option
 ASSIGN command 23

examples
 using the ADDRESS SET command 13
 using the ASKTIME command 19
 using the ASSIGN command 27
 using the BIF DEEDIT command 28
 using the CANCEL command 10
 using the DELAY command 64
 using the DELETE command 67
 using the DEQ command 72
 using the DUMP TRANSACTION command 77
 using the ENQ command 81
 using the ENTER TRACENUM command 84
 using the FORMATIME command 97
 using the FREEMAIN command 103
 using the GETMAIN command 124
 using the HANDLE ABEND command 129
 using the HANDLE AID command 131
 using the HANDLE CONDITION command 132
 using the LINK command 177
 using the LOAD command 179
 using the MONITOR command 182
 using the POST command 186
 using the READ command 195
 using the READQ TD command 205
 using the READQ TS command 207
 using the RELEASE command 232
 using the RETRIEVE command 238
 using the REWRITE command 243
 using the START ATTACH command 336
 using the START command 332
 using the WAIT EVENT command 351
 using the WAIT EXTERNAL command 353
 using the WAIT JOURNALNUM command 355
 using the WAITCICS command 359
 using the WRITE command 362
 using the WRITE JOURNALNUM command 364
 using the WRITEQ TD command 368
 using the XCTL command 372

EXCEPTION option
 ENTER TRACENUM command 83

exception support commands 7

exclusive control release, UNLOCK command 345

EXEC CICS command format 1

execution diagnostic facility (EDF) 335

exit from ASM program 384

exit, abnormal termination recovery 128

expiration time, notification when reached 185

EXPIRED condition
 DELAY command 64
 POST command 186
 WRITE OPERATOR command 366

EXPIRYTIME option
 VERIFY PASSWORD command 347

EXTATT operand
 DFHMDI 423
 DFHMSD 429

EXTDS option
 ASSIGN command 23

external security manager (ESM) 190, 335

EXTRACT ATTACH (LUTYPE6.1) command 85

EXTRACT ATTACH (MRO) command 87

EXTRACT ATTRIBUTES (APPC) command 89

EXTRACT ATTRIBUTES (MRO) command 90

EXTRACT LOGONMSG command 91

EXTRACT PROCESS command 92

EXTRACT TCT command 94

F

FACILITY option
 ASSIGN command 23

FCB option
 SPOOLOPEN REPORT command 311

FCI option
 ASSIGN command 23, 381

FCT option
 DUMP TRANSACTION command 75

field definition macro, BMS 411

field edit built-in function
See BIF DEEDIT

FIELD option
 BIF DEEDIT command 28

field separator operand 424, 430

FIELD value
 DFHMDI 423
 DFHMSD 429

FIELDS operand
 DFHMDI 424

file control
 commands 7
 deleting VSAM records 65
 end browse operation 78
 read next record 197
 read previous record 201
 release exclusive control 345
 specify start for browse 337
 update a record 242
 writing new record 360

FILE option
 DELETE command 65
 ENDBR command 78
 READ command 192
 READNEXT command 197
 READPREV command 201
 RESETBR command 234
 REWRITE command 242
 STARTBR command 337
 UNLOCK command 345
 WRITE command 360

filename
 definition 2, 3, 4

filename argument, CICS command format 2

FILENOTFOUND condition
 DELETE command 66
 ENDBR command 78
 READ command 194
 READNEXT command 199
 READPREV command 203
 RESETBR command 235
 REWRITE command 242
 STARTBR command 338
 UNLOCK command 345
 WRITE command 361

FIRST value
 DFHMDI 424

FLDSEP operand
 DFHMDI 424
 DFHMSD 430

FLENGTH option
 DUMP TRANSACTION command 75
 fullword alternative to LENGTH 387
 GDS RECEIVE command 117
 GDS SEND command 119
 GETMAIN command 123
 LOAD command 178
 RECEIVE (non-VTAM) command 225
 RECEIVE (VTAM) command 215
 SEND (non-VTAM) command 266
 SEND (VTAM) command 258
 SPOOLWRITE 324
 SPOOLWRITE MAPNAME command 326
 SPOOLWRITE REPORT command 329

FMH option
 CONVERSE (non-VTAM) command 60
 CONVERSE (VTAM) command 49
 SEND (non-VTAM) command 267
 SEND (VTAM) command 258
 START command 333

FMHPARM option
 SEND MAP command 274
 SEND PAGE command 278
 SEND TEXT command 283

FOLD operand
 DFHMSD 430

FOOT option
 SPOOLOPEN REPORT command 311

FOOTNUM option
 SPOOLOPEN REPORT command 311

FOR option
 DELAY command 63

FORMATTIME command 95

FORMFEED option
 SEND CONTROL command 270
 SEND MAP command 274
 SEND TEXT command 283

FORMS option
 SPOOLOPEN ESCAPE command 298
 SPOOLOPEN MAPNAME command 303
 SPOOLOPEN REPORT (SCS/T3270) command 317
 SPOOLOPEN REPORT command 311

FREE (APPC) command 99
 FREE (LUTYPE6.1) command 100
 FREE (MRO) command 101
 FREE command 98

free main storage 102

FREEKB option
 SEND CONTROL command 270
 SEND MAP command 274
 SEND TEXT command 283
 SEND TEXT NOEDIT command 289

FREEKB value
 DFHMDI 423
 DFHMSD 429

FREEMAIN command 102

FROM option
 CONVERSE (non-VTAM) command 60
 CONVERSE (VTAM) command 49
 DUMP TRANSACTION command 75
 ENTER TRACENUM command 83
 GDS SEND command 119
 ISSUE ADD command 139
 ISSUE PASS command 157
 ISSUE REPLACE command 164
 ISSUE SEND command 167
 RECEIVE MAP command 228
 REWRITE command 242
 SEND (non-VTAM) command 267
 SEND (VTAM) command 258
 SEND MAP command 274
 SEND TEXT command 283
 SEND TEXT MAPPED command 286
 SEND TEXT NOEDIT command 289
 SPOOLWRITE 324
 SPOOLWRITE MAPNAME command 326
 SPOOLWRITE REPORT command 329
 START ATTACH command 336
 START command 333
 WRITE command 360
 WRITE JOURNALNUM command 363
 WRITEQ TD command 367

FROM option (*continued*)
 WRITEQ TS command 369

FROMLENGTH option
 CONVERSE (non-VTAM) command 60
 CONVERSE (VTAM) command 49
 fullword alternative to FROMLENGTH 387

FROMLENGTH option
 CONVERSE (non-VTAM) command 60
 CONVERSE (VTAM) command 49
 ENTER TRACENUM command 83
 fullword length alternative (FROMLENGTH) 387

FRSET option
 SEND CONTROL command 270
 SEND MAP command 275

FRSET value
 DFHMDI 423
 DFHMSD 429

FSET value
 DFHMDF 416

Full Function Logical Unit, 3790 48, 214, 256

FULLDATE option
 FORMATTIME 96

fullword length option 387

FUNCERR condition
 ISSUE ABORT command 137
 ISSUE ADD command 140
 ISSUE END command 146
 ISSUE ERASE command 152
 ISSUE NOTE command 156
 ISSUE QUERY command 161
 ISSUE REPLACE command 165
 ISSUE SEND command 168
 ISSUE WAIT command 171

function management header
 See FMH

G

GCHARS option
 ASSIGN command 23

GCODES option
 ASSIGN command 23

GDS (generalized data stream) 6

GDS ALLOCATE command 104

GDS ASSIGN command 106

GDS CONNECT PROCESS command 107

GDS EXTRACT ATTRIBUTES command 109

GDS EXTRACT PROCESS command 110

GDS FREE command 111

GDS ISSUE ABEND command 112

GDS ISSUE CONFIRMATION command 113

GDS ISSUE ERROR command 114

GDS ISSUE PREPARE command 115

GDS ISSUE SIGNAL command 116

GDS RECEIVE command 117

GDS SEND command 119

GDS WAIT command 121

General Banking Terminal System (2980 General Banking Terminal System) 220

generalized data stream (GDS) 6

generic applid, XRF 21

GENERIC option
 DELETE command 65
 READ command 192
 RESETBR command 234
 STARTBR command 338

get main storage 122

GET NEXT IN PARENT, full form of GNP (DL/I) command 126

GET NEXT, full form of GN (DL/I) command 125

GET UNIQUE, full form of GU (DL/I) command 127

GETMAIN command 122

GINIT operand
 DFHMDF 416

GMMI option
 ASSIGN command 23

GN (DL/I) command 125

GNP (DL/I) command 126

GROUPID option
 SIGNON command 292

GRPNAME operand
 DFHMDF 416

GTEQ option
 READ command 192
 RESETBR command 234
 STARTBR command 338

GU (DL/I) command 127

H

HANDLE ABEND command 128

HANDLE AID command 130

HANDLE CONDITION command 132

HEAD option
 SPOOLOPEN REPORT command 311

HEADER operand
 DFHMDI 424

HEADER option
 SEND TEXT command 283

HEADNUM option
 SPOOLOPEN REPORT command 311

hhmss argument, CICS command format 2

HILIGHT operand
 DFHMDF 417
 DFHMDI 424
 DFHMSD 430

HILIGHT option
 ASSIGN command 23

HOLD option
 LOAD command 178
 SPOOLCLOSE REPORT command 295

HONEOM option
 SEND CONTROL command 270
 SEND MAP command 275
 SEND TEXT command 283
 SEND TEXT NOEDIT command 289
 host command processor LU, 3650/3680 254
 host conversational LU 3650
 (3270) 45, 253
 (3653) 46, 254
 HOURS option
 DELAY command 63
 POST command 186
 ROUTE command 244
 START command 333
 HTAB operand
 DFHMSD 430

I

IC value
 DFHMDF 416
 IGNORE CONDITION command 134
 IGREQCD condition
 CONVERSE (VTAM) command 51
 ISSUE SEND command 168
 SEND (VTAM) command 259
 SEND CONTROL command 271
 SEND MAP command 276
 SEND PAGE command 279
 SEND TEXT command 285
 SEND TEXT MAPPED command 287
 SEND TEXT NOEDIT command 290
 IGREQID condition
 SEND CONTROL command 271
 SEND MAP command 277
 SEND TEXT command 285
 SEND TEXT MAPPED command 287
 SEND TEXT NOEDIT command 290
 ILLOGIC condition
 DELETE command 66
 EIBRCODE 378
 ENDBR command 78
 READ command 194
 READNEXT command 199
 READPREV command 203
 RESETBR command 235
 REWRITE command 242
 SPOOLCLOSE REPORT command 296
 SPOOLOPEN ESCAPE command 300
 SPOOLOPEN INPUT command 301
 SPOOLOPEN MAPNAME command 305
 SPOOLOPEN OUTPUT command 308
 SPOOLOPEN REPORT (JCL) command 315
 SPOOLOPEN REPORT (SCS/T3270) command 320
 SPOOLOPEN REPORT command 314
 SPOOLOPEN RESUME command 321

ILLOGIC condition (*continued*)
 STARTBR command 338
 UNLOCK command 345
 WRITE command 361
 IMMEDIATE option
 RETURN command 240
 implicit SPOOLCLOSE 301
 INBFMH condition
 CONVERSE (non-VTAM) command 62
 CONVERSE (VTAM) command 51
 RECEIVE (non-VTAM) command 226
 RECEIVE (VTAM) command 216
 INITIAL operand
 DFHMDF 417
 initialize main storage 122
 initiate a task 333
 INITIMG option
 GETMAIN command 123
 INITPARM option
 ASSIGN command 23
 INITPARMLEN option
 ASSIGN command 23
 INPARTN option
 ASSIGN command 23
 RECEIVE MAP command 228
 input operation without data 390
 INPUT option
 SPOOLOPEN INPUT 301
 INPUTMSG option
 LINK command 175
 RETURN command 240
 XCTL command 371
 INPUTMSGLEN option
 LINK command 175
 RETURN command 240
 XCTL command 371
 INSERT, full form of ISRT (DL/I) command 135
 interactive logical units 47, 213, 255
 interface processor DFHEAI 384
 interpreter logical unit, 3650
 CONVERSE command 45
 ISSUE EODS command 150
 ISSUE LOAD command 155
 RECEIVE command 212
 SEND (VTAM) command 253
 interrogate a data set 161
 interval control
 ASKTIME options 19
 cancel interval control command 33
 CANCEL options 33
 commands 7
 DELAY options 63
 delay processing of task 63
 FORMATTIME options 95
 notification when specified time expires 185
 request current time of day 19

interval control (*continued*)

- retrieve data stored for task 237
- start a task 331
- wait for event to occur 350

INTERVAL option

- DELAY command 63
- POST command 186
- ROUTE command 244
- START command 333

INTO option

- CONVERSE (non-VTAM) command 60
- CONVERSE (VTAM) command 49
- EXTRACT LOGONMSG command 91
- GDS RECEIVE command 117
- ISSUE RECEIVE command 162
- READ command 192
- READNEXT command 197
- READPREV command 201
- READQ TD command 204
- READQ TS command 206
- RECEIVE (non-VTAM) command 225
- RECEIVE (VTAM) command 215
- RECEIVE MAP command 228
- RECEIVE PARTN command 230
- RETRIEVE command 237
- SPOOLREAD 323

INVALIDCOUNT option

- VERIFY PASSWORD command 347

INVERRTERM condition

- ROUTE command 246

INVITE option

- GDS SEND command 119
- SEND (non-VTAM) command 267
- SEND (VTAM) command 258

INVLDC condition

- ROUTE command 246
- SEND CONTROL command 271
- SEND MAP command 277
- SEND TEXT command 285

INVMPSTZ condition

- EIBRCODE byte 3 378
- RECEIVE MAP command 229
- SEND MAP command 277

INVOKINGPROG option

- ASSIGN command 23

INVPARTN condition

- RECEIVE MAP command 229
- RECEIVE PARTN command 231
- SEND CONTROL command 271
- SEND MAP command 277
- SEND TEXT command 285
- SEND TEXT NOEDIT command 290

INVPARTNSET condition

- SEND PARTNSET command 281

INVREQ condition

- ADDRESS command 12

INVREQ condition (*continued*)

- ALLOCATE (APPC) command 14
- ALLOCATE (LUTYPE6.1) command 16
- ALLOCATE (MRO) command 18
- ASSIGN command 27
- CHANGE PASSWORD command 35
- CHANGE TASK command 36
- CONNECT PROCESS command 39
- CONVERSE (VTAM) command 51
- DELAY command 64
- DELETE command 66
- DELETEQ TD command 69
- DELETEQ TS command 70
- DEQ command 71
- DUMP TRANSACTION command 77
- EIBRCODE bytes 1-3 378
- ENDBR command 78
- ENQ command 81
- ENTER TRACENUM command 83
- EXTRACT ATTACH (LUTYPE6.1) command 86
- EXTRACT ATTACH (MRO) command 88
- EXTRACT ATTRIBUTES (APPC) command 89
- EXTRACT PROCESS command 92
- EXTRACT TCT command 94
- FORMATTIME command 97
- FREE (APPC) command 99
- FREE (LUTYPE6.1) command 100
- FREE (MRO) command 101
- FREEMAIN command 102
- HANDLE AID command 131
- ISSUE ABEND command 136
- ISSUE ABORT command 137
- ISSUE ADD command 140
- ISSUE CONFIRMATION command 141
- ISSUE END command 146
- ISSUE ENDFILE command 148
- ISSUE ENDOUTPUT command 149
- ISSUE EODS command 150
- ISSUE ERASE command 152
- ISSUE ERASEAUP command 153
- ISSUE ERROR command 154
- ISSUE NOTE command 156
- ISSUE PASS command 157
- ISSUE PREPARE command 159
- ISSUE PRINT command 160
- ISSUE QUERY command 161
- ISSUE RECEIVE command 163
- ISSUE REPLACE command 165
- ISSUE SEND command 168
- ISSUE SIGNAL (APPC) command 169
- ISSUE WAIT command 171
- LINK command 176
- LOAD command 179
- MONITOR command 181
- POP HANDLE command 184
- POST command 186

INVREQ condition *(continued)*

PURGE MESSAGE command 188
 QUERY SECURITY command 191
 READ command 194
 READNEXT command 199
 READPREV command 203
 READQ TD command 205
 READQ TS command 207
 RECEIVE (non-VTAM) command 226
 RECEIVE MAP command 229
 RECEIVE PARTN command 231
 RELEASE command 232
 RESETBR command 235
 RETRIEVE command 238
 RETURN command 241
 REWRITE command 243
 ROUTE command 246
 SEND (non-VTAM) command 267
 SEND CONTROL command 271
 SEND MAP command 277
 SEND PAGE command 279
 SEND PARTNSET command 281
 SEND TEXT command 285
 SEND TEXT MAPPED command 287
 SEND TEXT NOEDIT command 290
 SIGNOFF command 291
 SIGNON command 293
 SPOOLCLOSE command 294
 SPOOLCLOSE REPORT command 296
 SPOOLOPEN ESCAPE command 300
 SPOOLOPEN INPUT command 301
 SPOOLOPEN MAPNAME command 305
 SPOOLOPEN OUTPUT command 308
 SPOOLOPEN REPORT (JCL) command 315
 SPOOLOPEN REPORT (SCS/T3270) command 320
 SPOOLOPEN REPORT command 314
 SPOOLOPEN RESUME command 321
 SPOOLREAD command 323
 SPOOLWRITE command 324
 SPOOLWRITE REPORT command 330
 START ATTACH command 336
 START command 335
 STARTBR command 339
 SYNCPOINT command 341
 SYNCPOINT ROLLBACK command 342
 UNLOCK command 345
 VERIFY PASSWORD command 347
 WAIT CONVID command 349
 WAIT EVENT command 350
 WAIT EXTERNAL command 353
 WAIT JOURNALNUM command 355
 WAIT TERMINAL command 357
 WAITCICS command 359
 WRITE command 361
 WRITE OPERATOR command 366
 WRITEQ TD command 367

INVREQ condition *(continued)*

WRITEQ TS command 370
 XCTL command 371

IOERR condition

DELETE command 67
 DUMP TRANSACTION command 77
 EIBRCODE 378
 ENDBR command 78
 READ command 194
 READNEXT command 199
 READPREV command 203
 READQ TD command 205
 READQ TS command 207
 RESETBR command 235
 RETRIEVE command 238
 REWRITE command 243
 SPOOLCLOSE command 294
 SPOOLCLOSE REPORT command 296
 SPOOLOPEN ESCAPE command 300
 SPOOLOPEN INPUT command 301
 SPOOLOPEN MAPNAME command 305
 SPOOLOPEN OUTPUT command 309
 SPOOLOPEN REPORT (JCL) command 316
 SPOOLOPEN REPORT (SCS/T3270) command 320
 SPOOLOPEN REPORT command 314
 SPOOLOPEN RESUME command 321
 SPOOLREAD command 323
 SPOOLWRITE command 324
 SPOOLWRITE MAPNAME command 327
 SPOOLWRITE REPORT command 330
 START command 335
 STARTBR command 339
 UNLOCK command 346
 WAIT JOURNALNUM command 355
 WRITE command 361
 WRITE JOURNALNUM command 364
 WRITEQ TD command 367
 WRITEQ TS command 370

ISCINVREQ condition

CANCEL command 33
 DELETE command 67
 DELETEQ TD command 69
 DELETEQ TS command 70
 ENDBR command 78
 READ command 195
 READNEXT command 200
 READPREV command 203
 READQ TD command 205
 READQ TS command 207
 RESETBR command 235
 REWRITE command 243
 START command 335
 STARTBR command 339
 UNLOCK command 346
 WRITE command 361
 WRITEQ TD command 367

ISCINVREQ condition (*continued*)
 WRITEQ TS command 370
 ISRT (DL/I) command 135
 ISSUE ABEND command 136
 ISSUE ABORT command 137
 ISSUE ADD command 139
 ISSUE CONFIRMATION command 141
 ISSUE COPY (3270 display) command 142
 ISSUE COPY (3270 logical) command 143
 ISSUE COPY command
 general information 390
 ISSUE DISCONNECT (default) command 144
 ISSUE DISCONNECT (LUTYPE6.1) command 145
 ISSUE DISCONNECT command
 general information 388
 ISSUE END command 146
 ISSUE ENDFILE command 148
 ISSUE ENDOUTPUT command 149
 ISSUE EODS command 150
 ISSUE ERASE command 151
 ISSUE ERASEAUP command 153
 general information 390
 ISSUE ERROR command 154
 ISSUE LOAD command 155
 ISSUE NOTE command 156
 ISSUE PASS command 157
 ISSUE PREPARE command 159
 ISSUE PRINT command 160
 general information 390
 ISSUE QUERY command 161
 ISSUE RECEIVE command 162
 ISSUE REPLACE command 164
 ISSUE RESET command 166
 ISSUE SEND command 167
 ISSUE SIGNAL (APPC) command 169
 ISSUE SIGNAL (LUTYPE6.1) command 170
 ISSUE SIGNAL command
 general information 388
 ISSUE WAIT command 171
 ITEM option
 READQ TS command 206
 WRITEQ TS command 369
 ITEMERR condition
 READQ TS command 207
 WRITEQ TS command 370
 IUTYPE option
 BUILD ATTACH (LUTYPE6.1) command 29
 BUILD ATTACH (MRO) command 31
 EXTRACT ATTACH (LUTYPE6.1) command 85
 EXTRACT ATTACH (MRO) command 87

J

JCL option
 SPOOLOPEN REPORT (JCL) command 315

JIDERR condition
 WAIT JOURNALNUM command 355
 WRITE JOURNALNUM command 364
 JOURNAL command 173
 journal control
 create a journal record 173
 journal record, creating 363
 journaling commands 7
 JOURNALNUM option
 WAIT JOURNALNUM command 355
 WRITE JOURNALNUM command 363
 JTYPEID option
 WRITE JOURNALNUM command 363
 JUSFIRST option
 SEND TEXT command 283
 JUSLAST option
 SEND TEXT command 283
 JUSTIFY operand
 DFHMDF 417
 DFHMDFI 424
 JUSTIFY option
 SEND TEXT command 283

K

katakana and English characters, mixed 60, 230
 KATAKANA option
 ASSIGN command 23
 katakana terminals
 CONVERSE (3270 logical) command 49
 CONVERSE (LUTYPE2/LUTYPE3) command 49
 CONVERSE command (3270 display) 60
 CONVERSE command (3600 BTAM) 60
 CONVERSE command (3735) 60
 CONVERSE command (3740) 60
 CONVERSE command (System/3) 60
 CONVERSE command (System/7) 60
 RECEIVE (3270 display) command 225
 RECEIVE (3270 logical) command 215
 RECEIVE (3790 3270-display) command 225
 RECEIVE (LUTYPE2/LUTYPE3) command 215
 RECEIVE (System/3) command 225
 RECEIVE (System/7) command 225
 RECEIVE MAP command 228
 RECEIVE PARTN command 230
 SEND (3600 BTAM) command 266
 SEND (3735) command 266
 SEND (3740) command 266
 SEND (System/3) command 266
 SEND (System/7) command 266
 KEEP option
 SPOOLCLOSE REPORT command 295
 KEEP or DELETE option
 SPOOLCLOSE 294
 KEYLENGTH option
 DELETE command 65

KEYLENGTH option (*continued*)
 ISSUE ERASE command 151
 ISSUE REPLACE command 164
 READ command 193
 READNEXT command 197
 READPREV command 201
 RESETBR command 234
 STARTBR command 338
 WRITE command 360
 KEYNUMBER option
 ISSUE ERASE command 151
 ISSUE REPLACE command 164
 keyword length 387

L

L40, L64, or L80 options
 SEND CONTROL command 270
 SEND MAP command 275
 SEND TEXT command 284
 SEND TEXT NOEDIT command 289
 label argument, CICS command format 2
 LABEL option
 HANDLE ABEND command 128
 LANG operand
 DFHMDI 424
 DFHMSD 430
 LANGINUSE option
 ASSIGN 24
 SIGNON command 292
 language codes 403
 LANGUAGECODE option
 SIGNON command 292
 LAST option
 GDS SEND command 119
 SEND (non-VTAM) command 267
 SEND (VTAM) command 258
 SEND CONTROL command 270
 SEND MAP command 275
 SEND PAGE command 278
 SEND TEXT command 284
 SEND TEXT MAPPED command 286
 SEND TEXT NOEDIT command 289
 LAST value
 DFHMDI 424
 LASTUSETIME option
 VERIFY PASSWORD command 347
 LDC operand
 DFHMSD 430
 LDC option
 CONVERSE (VTAM) command 49
 ROUTE command 244
 SEND (VTAM) command 258
 SEND CONTROL command 270
 SEND MAP command 275
 SEND TEXT command 284
 LDCMNEM option
 ASSIGN command 24
 LDCNUM option
 ASSIGN command 24
 LEAVEKB option
 CONVERSE (non-VTAM) command 60
 RECEIVE (non-VTAM) command 225
 SEND (non-VTAM) command 267
 LEFT value
 DFHMDF 417
 DFHMDI 424
 LENGERR condition
 BIF DEEDIT command 28
 CONNECT PROCESS command 39
 CONVERSE (non-VTAM) command 62
 CONVERSE (VTAM) command 51
 DEQ command 71
 EIBRCODE byte 1 378
 ENQ command 81
 ENTER TRACENUM command 84
 EXTRACT PROCESS command 93
 GETMAIN command 124
 ISSUE COPY (3270 logical) command 143
 ISSUE PASS command 157
 ISSUE RECEIVE command 163
 LINK command 176
 LOAD command 179
 QUERY SECURITY command 191
 READ command 195
 READNEXT command 200
 READPREV command 203
 READQ TD command 205
 READQ TS command 207
 RECEIVE (non-VTAM) command 226
 RECEIVE (VTAM) command 216
 RECEIVE PARTN command 231
 RETRIEVE command 238
 RETURN command 241
 REWRITE command 243
 SEND (non-VTAM) command 267
 SEND (VTAM) command 259
 SEND TEXT command 285
 SPOOLREAD command 323
 SPOOLWRITE command 324
 SPOOLWRITE MAPNAME command 327
 SPOOLWRITE REPORT command 330
 START ATTACH command 336
 START command 335
 WRITE command 362
 WRITE JOURNALNUM command 364
 WRITE OPERATOR command 366
 WRITEQ TD command 367
 WRITEQ TS command 370
 XCTL command 371
 LENGTH operand
 DFHMDF 417

- LENGTH option
 - BIF DEEDIT command 28
 - built-in function 28
 - default (assembler language) 4
 - default (C) 3
 - default (PL/I) 3
 - DEQ command 71
 - DUMP TRANSACTION command 75
 - ENQ command 80
 - EXTRACT LOGONMSG command 91
 - fullword length alternative (FLENGTH) 387
 - GETMAIN command 123
 - ISSUE ADD command 139
 - ISSUE PASS command 157
 - ISSUE RECEIVE command 162
 - ISSUE REPLACE command 164
 - ISSUE SEND command 167
 - LINK command 175
 - LOAD command 178
 - READ command 193
 - READNEXT command 198
 - READPREV command 201
 - READQ TD command 204
 - READQ TS command 206
 - RECEIVE (non-VTAM) command 225
 - RECEIVE (VTAM) command 215
 - RECEIVE MAP command 228
 - RECEIVE PARTN command 230
 - RETRIEVE command 237
 - RETURN command 241
 - REWRITE command 242
 - SEND (non-VTAM) command 267
 - SEND (VTAM) command 259
 - SEND MAP command 275
 - SEND TEXT command 284
 - SEND TEXT MAPPED command 286
 - SEND TEXT NOEDIT command 289
 - START ATTACH command 336
 - START command 333
 - WRITE command 360
 - WRITE JOURNALNUM command 363
 - WRITEQ TD command 367
 - WRITEQ TS command 369
 - XCTL command 371
- LENGTH value
 - DFHMDI 422
 - DFHMSD 429
- LENGTHLIST option
 - DUMP TRANSACTION command 75
- LIGHTPEN option
 - HANDLE AID command 130
- LINE operand
 - DFHMDI 424
- LINE option
 - SPOOLWRITE 324
- line value
 - DFHMDI 425
- line,column value
 - DFHMDF 419
- LINEADDR option
 - CONVERSE (non-VTAM) command 60
 - SEND (non-VTAM) command 267
- LINELENGTH option
 - SPOOLOPEN REPORT (SCS/T3270) command 317
 - SPOOLOPEN REPORT command 311
- LINES option
 - SPOOLOPEN REPORT (SCS/T3270) command 318
 - SPOOLOPEN REPORT command 312
- LINK command 174
- link to program expecting return 174
- LIST option
 - ROUTE command 245
- literal constants 4
- LLID option
 - GDS RECEIVE command 117
- LOAD (DL/I) command 180
- LOAD command 178
- load programs, tables, or maps 178
- LOADING condition
 - DELETE command 67
 - READ command 195
 - STARTBR command 339
 - WRITE command 362
- LOG option 315, 318
 - SPOOLCLOSE REPORT command 295
 - SPOOLOPEN ESCAPE command 299
 - SPOOLOPEN MAPNAME command 303
 - SPOOLOPEN REPORT (JCL) command 315
 - SPOOLOPEN REPORT (SCS/T3270) command 318
 - SPOOLOPEN REPORT command 312
 - SPOOLWRITE MAPNAME command 326
 - SPOOLWRITE REPORT command 329
- logical device code (LDC) 43, 252
- logical messages, BMS
 - completing a logical message 278
 - full BMS
 - ROUTE command 244
 - purging a logical message 188
 - routing a logical message 244
- LOGICAL option
 - SPOOLOPEN ESCAPE command 299
 - SPOOLOPEN MAPNAME command 303
 - SPOOLOPEN REPORT (SCS/T3270) command 318
 - SPOOLOPEN REPORT command 312
- LOGMESSAGE option
 - QUERY SECURITY command 190
- LOGMODE option
 - ISSUE PASS command 157
- LOGONLOGMODE option
 - ISSUE PASS 157

- LU (logical unit)
 - 3270 Information Display System 43, 143, 210, 251
 - 3270 SCS Printer 42, 250
 - 3270-Display, LUTYPE2 41, 209, 249
 - 3270-Display, LUTYPE3 209, 249
 - 3600 (3601) 43, 211, 252
 - 3600 (3614) 44, 212, 252
 - 3600 pipeline 211, 251
 - 3650 host conversational (3270) 45, 253
 - 3650 host conversational (3653) 46, 254
 - 3650 interpreter 45, 150, 155, 212, 253
 - 3650/3680 host command processor 254
 - 3770 batch 47, 214, 255
 - 3790 (3270-display) 48, 224, 257
 - 3790 (3270-printer) 257
 - 3790 full-function 214, 256
 - 3790 full-function or inquiry 48
 - 3790 SCS printer 256
 - batch 47, 214, 255
 - conversing with (CONVERSE) 388
 - interactive 47
 - reading data from 162, 387
 - writing data to 139, 387
- LUNAME option
 - ISSUE PASS command 157
- LUTYPE2, 3270-Display LU 41, 209, 249
- LUTYPE3, 3270-Display LU 209, 249
- LUTYPE4
 - logical unit 41, 209, 249
- LUTYPE6.1 logical unit
 - acquiring a session 16
 - communicating on LUTYPE6.1 session 42
 - converting 8-character names to 4 characters 94
 - disconnecting 145
 - ensuring terminal operation has completed 357
 - getting information about 183
 - receiving data 210
 - requesting change of direction 170
 - retrieving values from an LUTYPE6.1 header 85
 - sending data 250
 - specifying values for an MRO attach header 31
 - specifying values for LUTYPE6.1 attach header 29

M

- macros, BMS, summary 411
- magnetic slot reader (MSR) 407
- MAIN option
 - WRITEQ TS command 369
- main storage 122
- map definition macro, BMS 411, 422
- MAP option
 - RECEIVE MAP command 229
 - SEND MAP command 275
- MAP value
 - DFHMSD 432

- MAPATTS operand
 - DFHMMDI 425
 - DFHMSD 430
- MAPCOLUMN option
 - ASSIGN command 24
- MAPERROR condition
 - SPOOLWRITE MAPNAME command 327
- MAPFAIL condition
 - RECEIVE MAP command 229
- MAPHEIGHT option
 - ASSIGN command 24
- MAPLINE option
 - ASSIGN command 24
- MAPNAME option
 - SPOOLOPEN MAPNAME command 304
 - SPOOLWRITE MAPNAME command 326
- MAPONLY option
 - SEND MAP command 275
 - SPOOLWRITE MAPNAME command 327
- MAPONLY value
 - DFHMMDI 424
 - DFHMSD 430
- maps, loading 178
- mapset definition macro (DFHMSD) 411, 428
- MAPSET option
 - RECEIVE MAP command 229
 - SEND MAP command 275
 - SPOOLOPEN MAPNAME command 304
 - SPOOLWRITE MAPNAME command 327
- MAPSFX operand
 - DFHPDI 433
- MAPWIDTH option
 - ASSIGN command 24
- MASSINSERT option
 - WRITE command 360
- MAXLENGTH option
 - CONVERSE (non-VTAM) command 60
 - CONVERSE (VTAM) command 49
 - fullword alternative to MAXLENGTH 387
 - GDS RECEIVE command 117
 - RECEIVE (non-VTAM) command 225
 - RECEIVE (VTAM) command 215
 - SPOOLREAD 323
- MAXLENGTH option
 - CONVERSE (non-VTAM) command 60
 - CONVERSE (VTAM) command 49
 - fullword length alternative (MAXLENGTH) 387
 - RECEIVE (non-VTAM) command 225
 - RECEIVE (VTAM) command 215
 - WRITE OPERATOR command 365
- MAXLIFETIME option
 - DEQ command 71
 - ENQ command 80
- MAXPROCLEN option
 - EXTRACT PROCESS command 92
 - GDS EXTRACT PROCESS command 110

MCC option
 SPOOLOPEN OUTPUT 307
 SPOOLOPEN REPORT command 312

MINUTES option
 DELAY command 63
 POST command 186
 ROUTE command 245
 START command 334

MMDDYY option
 FORMATTIME command 96

MMDDYYYY option
 FORMATTIME command 96

MODE operand
 DFHMDI 425
 DFHMSD 430

model codes (terminal) 381

MODENAME option
 GDS ALLOCATE command 104

MONITOR command 181

monitoring application performance 181

monitoring commands 7

MONTHOFYEAR option
 FORMATTIME command 96

MSR (magnetic slot reader)
 control byte values and constants 407
 DFHMSRCA, 407

MSR option
 SEND CONTROL command 270
 SEND MAP command 275
 SEND TEXT command 284

MSRCONTROL option
 ASSIGN command 24

multi region operation (MRO) commands
 ALLOCATE 18
 BUILD ATTACH 31
 CONVERSE command 52
 EXTRACT ATTACH 87
 EXTRACT ATTRIBUTES 90
 FREE 101
 RECEIVE 217
 SEND 261

multiple base registers 385

MUSTENTER value
 DFHMDI 419
 DFHMDI 426
 DFHMSD 432

MUSTFILL value
 DFHMDI 419
 DFHMDI 426
 DFHMSD 432

N

name argument, CICS command format 2

NAME option
 WAIT EVENT command 350

NAME option (*continued*)
 WAIT EXTERNAL command 352
 WAITCICS command 358

NAMEERROR condition
 SPOOLCLOSE REPORT command 296
 SPOOLOPEN REPORT command 314
 SPOOLOPEN RESUME command 322
 SPOOLWRITE REPORT command 330

naming restriction 383

national language codes 403

NATLANG option
 SIGNON command 293

NATLANGINUSE option
 ASSIGN command 24
 SIGNON command 293

NETNAME option
 ASSIGN command 24
 EXTRACT TCT command 94

NETNAMEIDERR condition
 ALLOCATE (APPC) command 14

new tasks, passing data to 333

NEWPASSWORD option
 CHANGE PASSWORD command 35
 SIGNON command 293

NEXT option
 READQ TS command 206

NEXT value
 DFHMDI 422, 424

NEXTTRANSID option
 ASSIGN command 24

NLEOM option
 ROUTE command 245
 SEND MAP command 276
 SEND TEXT command 284

NO value
 DFHMDI 423, 425
 DFHMSD 430, 432

NOAUTOPAGE option
 SEND PAGE command 278

NOCC option
 SPOOLOPEN OUTPUT 307
 SPOOLOPEN REPORT command 312

NOCC or ASA or MCC option
 SPOOLOPEN OUTPUT 307

NOCHECK option
 START command 334

NOCONV option
 SPOOLOPEN REPORT (SCS/T3270) command 318

NODE option
 SPOOLOPEN ESCAPE command 299
 SPOOLOPEN MAPNAME command 304
 SPOOLOPEN OUTPUT 307
 SPOOLOPEN REPORT (SCS/T3270) command 318
 SPOOLOPEN REPORT command 312

NODUMP option
 ABEND command 10

NOFLUSH option
 SEND MAP command 276

NOHANDLE option
 deactivating HANDLE CONDITION command 132
 option 5
 overriding HANDLE AID 6

NOJBUFSP condition
 WRITE JOURNALNUM command 364

NONUM option
 SPOOLOPEN REPORT command 312

NONVAL condition
 ISSUE LOAD command 155

NOPASSBKRD condition
 RECEIVE (non-VTAM) command 226

NOPASSBKWR condition
 SEND (non-VTAM) command 267

NOQUEUE option
 ALLOCATE (APPC) command 14
 ALLOCATE (LUTYPE6.1) command 16
 ALLOCATE (MRO) command 18
 GDS ALLOCATE command 104

NOQUIESCE
 ISSUE PASS command 157

NORM value
 DFHMDF 416

NOSEP option
 SPOOLOPEN ESCAPE command 299
 SPOOLOPEN MAPNAME command 304
 SPOOLOPEN REPORT (SCS/T3270) command 318
 SPOOLOPEN REPORT command 312

NOSPACE condition
 DUMP TRANSACTION command 77
 REWRITE command 243
 SPOOLCLOSE REPORT command 296
 SPOOLOPEN ESCAPE command 300
 SPOOLOPEN INPUT command 302
 SPOOLOPEN MAPNAME command 305
 SPOOLOPEN OUTPUT command 309
 SPOOLOPEN REPORT (JCL) command 316
 SPOOLOPEN REPORT (SCS/T3270) command 320
 SPOOLOPEN REPORT command 314
 SPOOLOPEN RESUME command 322
 SPOOLWRITE command 324
 SPOOLWRITE MAPNAME command 328
 SPOOLWRITE REPORT command 330
 WRITE command 362
 WRITEQ TD command 367
 WRITEQ TS command 370

NOSPOOL condition
 SPOOLCLOSE command 294
 SPOOLCLOSE REPORT command 297
 SPOOLOPEN ESCAPE command 300
 SPOOLOPEN INPUT command 302
 SPOOLOPEN MAPNAME command 305
 SPOOLOPEN OUTPUT command 309
 SPOOLOPEN REPORT (JCL) command 316

NOSPOOL condition (*continued*)
 SPOOLOPEN REPORT (SCS/T3270) command 320
 SPOOLOPEN REPORT command 314
 SPOOLOPEN RESUME command 322
 SPOOLREAD command 323
 SPOOLWRITE command 325
 SPOOLWRITE MAPNAME command 328
 SPOOLWRITE REPORT command 330

NOSTART condition
 ISSUE LOAD command 155

NOSTG condition
 DUMP TRANSACTION command 77
 GETMAIN command 124

NOSUSPEND option
 ALLOCATE (APPC) 14
 ALLOCATE (LUTYPE6.1) command 16
 ENQ command 81
 GETMAIN command 123
 READQ TD command 204
 WRITE JOURNALNUM command 363
 WRITEQ TS command 369

NOTALLOC condition
 CONNECT PROCESS command 39
 CONVERSE (non-VTAM) command 62
 CONVERSE (VTAM) command 51
 EXTRACT ATTACH (LUTYPE6.1) command 86
 EXTRACT ATTACH (MRO) command 88
 EXTRACT ATTRIBUTES (APPC) command 89
 EXTRACT ATTRIBUTES (MRO) command 90
 EXTRACT LOGONMSG command 91
 EXTRACT PROCESS command 93
 EXTRACT TCT command 94
 FREE (APPC) command 99
 FREE (LUTYPE6.1) command 100
 FREE (MRO) command 101
 FREE command 98
 ISSUE ABEND command 136
 ISSUE CONFIRMATION command 141
 ISSUE COPY (3270 display) command 142
 ISSUE COPY (3270 logical) command 143
 ISSUE DISCONNECT (LUTYPE6.1) command 145
 ISSUE ENDFILE command 148
 ISSUE ENDOUTPUT command 149
 ISSUE EODS command 150
 ISSUE ERASEAUP command 153
 ISSUE ERROR command 154
 ISSUE LOAD command 155
 ISSUE PASS command 158
 ISSUE PREPARE command 159
 ISSUE PRINT command 160
 ISSUE SIGNAL (APPC) command 169
 ISSUE SIGNAL (LUTYPE6.1) command 170
 POINT command 183
 RECEIVE (non-VTAM) command 226
 RECEIVE (VTAM) command 216
 SEND (non-VTAM) command 267

NOTALLOC condition (*continued*)
 SEND (VTAM) command 259
 WAIT CONVID command 349
 WAIT SIGNAL command 356
 WAIT TERMINAL command 357
 NOTAUTH condition
 CANCEL command 33
 CHANGE PASSWORD command 35
 DELETE command 67
 DELETEQ TD command 69
 DELETEQ TS command 70
 ENDBR command 78
 HANDLE ABEND command 128
 LINK command 176
 LOAD command 179
 READ command 195
 READNEXT command 200
 READPREV command 203
 READQ TD command 205
 READQ TS command 207
 RELEASE command 232
 RESETBR command 236
 REWRITE command 243
 SIGNON command 293
 SPOOLCLOSE REPORT command 297
 SPOOLOPEN INPUT command 302
 SPOOLOPEN RESUME command 322
 SPOOLWRITE MAPNAME command 328
 SPOOLWRITE REPORT command 330
 START ATTACH command 336
 START command 335
 STARTBR command 339
 UNLOCK command 346
 VERIFY PASSWORD command 348
 WRITE command 362
 WRITE JOURNALNUM command 364
 WRITEQ TD command 367
 WRITEQ TS command 370
 XCTL command 371
 NOTFND condition
 CANCEL command 34
 DELETE command 67
 QUERY SECURITY command 191
 READ command 195
 READNEXT command 200
 READPREV command 203
 RESETBR command 236
 RETRIEVE command 238
 REWRITE 243
 SPOOLOPEN INPUT command 302
 STARTBR command 339
 NOTOPEN condition
 DELETE command 67
 DUMP TRANSACTION command 77
 READ command 195
 READQ TD command 205

NOTOPEN condition (*continued*)
 STARTBR command 339
 UNLOCK command 346
 WAIT JOURNALNUM command 355
 WRITE command 362
 WRITE JOURNALNUM command 364
 WRITEQ TD command 368
 NOTTRUNCATE option
 CONVERSE (non-VTAM) command 61
 CONVERSE (VTAM) command 50
 RECEIVE (non-VTAM) command 225
 RECEIVE (VTAM) command 215
 NOWAIT option
 ISSUE ADD command 139
 ISSUE ERASE command 151
 ISSUE REPLACE command 164
 ISSUE SEND command 167
 NUM value
 DFHMDF 416
 number value
 DFHMDF 419
 DFHMDI 422, 424
 NUMEVENTS option
 WAIT EXTERNAL command 352
 WAITCICS command 358
 NUMITEMS option
 READQ TS command 206
 WRITEQ TS command 370
 NUMREC option
 DELETE command 65
 ISSUE ADD command 139
 ISSUE ERASE command 151
 ISSUE REPLACE command 164
 NUMROUTES option
 WRITE OPERATOR command 365
 NUMSEGMENTS option
 DUMP TRANSACTION command 76
 NUMTAB option
 ASSIGN command 24

O

OBFMT operand
 DFHMDI 425
 DFHMSD 430
 OCCURS operand
 DFHMDF 417
 OFF value
 DFHMDF 417
 DFHMDI 424
 DFHMSD 430
 OIDCARD option
 SIGNON command 293
 OPCLASS option
 ASSIGN command 24
 ROUTE command 245

- OPENERR condition
 - DUMP TRANSACTION command 77
- OPERID option
 - HANDLE AID command 130
- OPERKEYS option
 - ASSIGN command 24
- OPERPURGE option
 - SEND PAGE command 279
- OPID option
 - ASSIGN command 24
- OPSECURITY option
 - ASSIGN command 24
- options
 - BMS 228, 282
 - length 387
- OPTIONS(MAIN)
 - in PL/I 383
- ORGABCODE option
 - ASSIGN command 24
- OUTDESCR option
 - SPOOL OPEN OUTPUT 308
 - SPOOL OPEN REPORT command 312
- OUTDESCRERR condition
 - SPOOL OPEN OUTPUT command 309
- OUTLINE operand
 - DFHMDF 418
 - DFHMDI 425
 - DFHMSD 430
- OUTLINE option
 - ASSIGN command 24
- OUTPARTN option
 - SEND CONTROL command 271
 - SEND MAP command 276
 - SEND TEXT command 284
 - SEND TEXT NOEDIT command 289
- output control, 2980 General Banking Terminal System 220
- OUTPUT option
 - SPOOL OPEN OUTPUT 308
- output to common buffer, 2980 221
- OVERFLOW condition
 - SEND MAP command 277

P

- PA1-PA3 option
 - HANDLE AID command 130
- PAGE option
 - SPOOLWRITE 324
- PAGENUM option
 - ASSIGN command 24
- PAGING option
 - SEND CONTROL command 271
 - SEND MAP command 276
 - SEND TEXT command 284
 - SEND TEXT MAPPED command 286
 - SEND TEXT NOEDIT command 289
- partition definition macro (DFHPDI) 412, 433
- partition set definition macro (DFHPSD) 412, 434
- PARTN operand
 - DFHMDI 425
 - DFHMSD 430
- PARTN option
 - RECEIVE PARTN command 231
- PARTNER option
 - ALLOCATE (APPC) command 14
 - CONNECT PROCESS command 38
 - GDS ALLOCATE command 104
 - GDS CONNECT PROCESS command 107
- PARTNERIDERR condition
 - ALLOCATE (APPC) command 15
 - CONNECT PROCESS command 39
- PARTNFAIL condition
 - RECEIVE MAP command 229
- PARTNPAGE option
 - ASSIGN command 25
- PARTNS option
 - ASSIGN command 25
- PARTNSET option
 - ASSIGN command 25
- PASSBK option
 - RECEIVE (non-VTAM) command 225
 - SEND (non-VTAM) command 267
- passbook control, 2980 220
- passing a session 157
- passing control
 - expecting return (LINK) 174
 - on receipt of an AID (HANDLE AID command) 130
 - on receipt of an AID (IGNORE AID) 134
 - without return (XCTL) 371
- passing data to new tasks 333
- PASSWORD option
 - CHANGE PASSWORD command 35
 - SIGNON command 293
 - VERIFY PASSWORD command 347
- PCT option
 - DUMP TRANSACTION command 76
- performance, application, monitoring 181
- PF1-24 option
 - HANDLE AID command 130
- PFXLENG option
 - WRITE JOURNALNUM command 363
- PGMIDERR condition
 - HANDLE ABEND command 129
 - LINK command 176
 - LOAD command 179
 - RELEASE command 232
 - XCTL command 372
- PHYSICAL option
 - SPOOL OPEN ESCAPE command 299
 - SPOOL OPEN MAPNAME command 304
 - SPOOL OPEN REPORT (SCS/T3270) command 318
 - SPOOL OPEN REPORT command 312

PICIN operand
 DFHMDF 418
 PICOUT operand
 DFHMDF 418
 pipeline logical units 211, 251
 PIPELENGTH option
 CONNECT PROCESS command 38
 EXTRACT PROCESS command 92
 GDS CONNECT PROCESS command 107
 GDS EXTRACT PROCESS command 110
 PIPLIST option
 CONNECT PROCESS command 38
 EXTRACT PROCESS command 92
 GDS CONNECT PROCESS command 107
 GDS EXTRACT PROCESS command 110
 PL/I language
 argument values 3
 LENGTH option default 3
 PROCEDURE statement 383
 translated code 383
 POINT command 183
 POINT option
 MONITOR command 181
 pointer-ref argument, CICS command format 2
 pointer-value argument, CICS command format 2
 POP HANDLE command 184
 POS operand 413
 DFHMDF 419
 POST command 185
 posting timer-event control area 185
 PPT option
 DUMP TRANSACTION command 76
 PREFIX option
 WRITE JOURNALNUM command 363
 PRINCONVID option
 GDS ASSIGN command 106
 PRINSYSID option
 ASSIGN command 25
 GDS ASSIGN command 106
 print displayed information 390
 PRINT option
 ISSUE ABORT command 137
 ISSUE END command 146
 ISSUE SEND command 167
 ISSUE WAIT command 171
 SEND CONTROL command 271
 SEND MAP command 276
 SEND TEXT command 284
 SEND TEXT NOEDIT command 289
 SPOOLOPEN INPUT 301
 SPOOLOPEN OUTPUT 308
 PRINT value
 DFHMDI 422
 DFHMSD 428
 printer control character list, DFHBMSCA 405
 printer spooling options
 ASA 310
 BOTHNUM 310
 CLASS 298, 303, 310, 317
 COPIES 298, 303, 311, 317
 DATETIME 311
 DELETE 295
 DESTINATION 298, 303, 311, 317
 DISP 295
 ESCAPE 298
 FCB 311
 FLENGTH 326, 329
 FOOT 311
 FOOTNUM 311
 FORMS 298, 303, 311, 317
 FROM 326, 329
 HEAD 311
 HEADNUM 311
 HOLD 295
 JCL 315
 KEEP 295
 LINELENGTH 311, 317
 LINES 312, 318
 LOG 295, 299, 303, 312, 315, 318, 326, 329
 LOGICAL 299, 303, 312, 318
 MAPNAME 304, 326
 MAPONLY 327
 MAPSET 304, 327
 MCC 312
 NOCC 312
 NOCONV 318
 NODE 299, 304, 312, 318
 NONUM 312
 NOSEP 299, 304, 312, 318
 PHYSICAL 299, 304, 312, 318
 PRINTFAIL 299, 304, 312, 318
 PRIORITY 299, 304, 313, 318
 PWRSYSID 299, 304, 313, 318
 RELEASE 296
 REPORT 296, 299, 304, 313, 315, 319, 321, 327, 329
 RESUME 296, 321
 RSL 299, 304, 313, 319
 SCS 319
 SEP 299, 305, 313, 319
 SOSI1 313, 319
 SOSI2 313, 319
 T3270 319
 TITLE 300, 305, 313, 319
 TOKEN 296, 300, 305, 313, 315, 319, 321, 327, 330
 USERDATA 300, 305, 313, 319
 USERID 300, 305, 313, 320
 PRINTFAIL option
 SPOOLOPEN ESCAPE command 299
 SPOOLOPEN MAPNAME command 304
 SPOOLOPEN REPORT (SCS/T3270) command 318
 SPOOLOPEN REPORT command 312

priority of task, changing 36
 PRIORITY option
 CHANGE TASK command 36
 SPOOLOPEN ESCAPE command 299
 SPOOLOPEN MAPNAME command 304
 SPOOLOPEN REPORT (SCS/T3270) command 318
 SPOOLOPEN REPORT command 313
 PROCESS option
 BUILD ATTACH (LUTYPE6.1) command 30
 BUILD ATTACH (MRO) command 32
 EXTRACT ATTACH (LUTYPE6.1) command 85
 EXTRACT ATTACH (MRO) command 87
 processing task, control delay of 63
 PROCLENGTH option
 CONNECT PROCESS command 38
 EXTRACT PROCESS command 92
 GDS CONNECT PROCESS command 107
 GDS EXTRACT PROCESS command 110
 PROCNAME option
 CONNECT PROCESS command 38
 EXTRACT PROCESS command 92
 GDS CONNECT PROCESS command 107
 GDS EXTRACT PROCESS command 110
 PROFILE option
 ALLOCATE (APPC) command 14
 ALLOCATE (LUTYPE6.1) command 16
 program control
 commands 7
 deleting loaded program 232
 LINK command options 175
 linking to another program 174
 load a program, table, or map 178
 returning program control 240
 transfer program control 371
 PROGRAM option
 ASSIGN command 25
 DUMP TRANSACTION command 76
 HANDLE ABEND command 128
 ISSUE LOAD command 155
 LINK command 175
 LOAD command 178
 RELEASE command 232
 XCTL command 371
 PROT value
 DFHMDF 416
 PROTECT option
 START command 334
 PS operand
 DFHMDF 419
 DFHMDI 425
 DFHMSD 431
 PS option
 ASSIGN command 25
 PSEUDOBIN option
 CONVERSE (non-VTAM) command 61
 RECEIVE (non-VTAM) command 226

PSEUDOBIN option (*continued*)
 SEND (non-VTAM) command 267
 psid value
 DFHMDF 419
 DFHMDI 425
 DFHMSD 431
 PUNCH option
 SPOOLOPEN INPUT 301
 SPOOLOPEN OUTPUT 308
 PURGE MESSAGE command 188
 PURGEABILITY option
 WAIT EXTERNAL command 352
 WAITCICS command 358
 PUSH HANDLE command 189
 PWRSYSID option
 SPOOLOPEN ESCAPE command 299, 304, 313, 318

Q

QBUSY condition
 READQ TD command 205
 QIDERR condition
 DELETEQ TD command 69
 DELETEQ TS command 70
 QUERY SECURITY command 191
 READQ TD command 205
 READQ TS command 207
 WRITEQ TD command 368
 WRITEQ TS command 370
 QNAME option
 ASSIGN command 25
 QUERY SECURITY command 190
 QUEUE option
 BUILD ATTACH (LUTYPE6.1) command 30
 BUILD ATTACH (MRO) command 32
 DELETEQ TD command 69
 DELETEQ TS command 70
 EXTRACT ATTACH (LUTYPE6.1) command 86
 EXTRACT ATTACH (MRO) command 88
 READQ TD command 204
 READQ TS command 206
 RETRIEVE command 237
 START command 334
 WRITEQ TD command 367
 WRITEQ TS command 370
 QZERO condition
 READQ TD command 205

R

RBA option
 DELETE command 66
 READ command 193
 READNEXT command 198
 READPREV command 202
 RESETBR command 235

RBA option (*continued*)
 STARTBR command 338
 WRITE command 361

RDATT condition
 CONVERSE (non-VTAM) command 62
 RECEIVE (non-VTAM) command 226
 RECEIVE MAP command 229

reactivate an ABEND exit 128

read attention 55

READ command 192

READ option
 QUERY SECURITY command 190

reading records
 batch data interchange 162
 browsing, next 197
 browsing, previous (VSAM) 201
 file control 192
 from temporary storage queue 206
 from terminal or LU 387
 from transient data queue 204

READNEXT command 197

READPREV command 201

READQ TD command 204

READQ TS command 206

RECEIVE (2260) command 219

RECEIVE (2741) command 219

RECEIVE (2980) command 220

RECEIVE (3270 display) command 222

RECEIVE (3270 logical) command 210

RECEIVE (3600 BTAM) command 222

RECEIVE (3600 pipeline) command 211

RECEIVE (3600-3601) command 211

RECEIVE (3600-3614) command 212

RECEIVE (3650) command 212

RECEIVE (3735) command 223

RECEIVE (3740) command 223

RECEIVE (3767) command 213

RECEIVE (3770) command 214

RECEIVE (3790 3270-display) command 224

RECEIVE (3790 full-function or inquiry) command 214

RECEIVE (APPC) command 208

RECEIVE (LUTYPE2/LUTYPE3) command 209

RECEIVE (LUTYPE4) command 209

RECEIVE (LUTYPE6.1) command 210

RECEIVE (MRO) command 217

RECEIVE (non-VTAM) command 217

RECEIVE (System/3) command 218

RECEIVE (System/7) command 218

RECEIVE (VTAM default) command 208

RECEIVE command
 input operation without data 390
 read from terminal or logical unit 387

RECEIVE MAP command 228

RECEIVE PARTN command 230

RECFM option
 BUILD ATTACH (LUTYPE6.1) command 30

RECFM option (*continued*)
 BUILD ATTACH (MRO) command 32
 EXTRACT ATTACH (LUTYPE6.1) command 86
 EXTRACT ATTACH (MRO) command 88

records
 deleting VSAM 65
 reading 162, 192
 release exclusive control 345
 requesting next number 156
 updating 164, 242
 writing new 360
 writing new (adding) 139

register contents in assembler language 384

relative byte address (RBA) 66

RELEASE command 232

RELEASE option
 SEND PAGE command 279
 SPOOLCLOSE REPORT command 296

relocatable expression 4

REPL (DL/I) command 233

REPLACE, full form of REPL (DL/I) command 233

REPLY option
 WRITE OPERATOR command 365

REPLYLENGTH option
 WRITE OPERATOR command 366

report controller
 commands 8

REPORT option
 SPOOLCLOSE REPORT command 296
 SPOOLOPEN ESCAPE command 299
 SPOOLOPEN MAPNAME command 304
 SPOOLOPEN REPORT (JCL) command 315
 SPOOLOPEN REPORT (SCS/T3270) command 319
 SPOOLOPEN REPORT command 313
 SPOOLOPEN RESUME command 321
 SPOOLWRITE MAPNAME command 327
 SPOOLWRITE REPORT command 329

REQID option
 CANCEL command 33
 DELAY command 63
 ENDBR command 78
 POST command 186
 READNEXT command 198
 READPREV command 202
 RESETBR command 235
 ROUTE command 245
 SEND CONTROL command 271
 SEND MAP command 276
 SEND TEXT command 284
 SEND TEXT MAPPED command 286
 SEND TEXT NOEDIT command 289
 START command 334
 STARTBR command 338
 WAIT JOURNALNUM command 355
 WRITE JOURNALNUM command 363

RESCLASS option
 QUERY SECURITY command 190
 RESET option
 HANDLE ABEND command 128
 reset start for browse 234
 RESETBR command 234
 RESID option
 QUERY SECURITY command 190
 RESIDLENGTH option
 QUERY SECURITY command 190
 RESOURCE option
 BUILD ATTACH (LUTYPE6.1) command 30
 BUILD ATTACH (MRO) command 32
 DEQ command 71
 ENQ command 81
 ENTER TRACENUM command 83
 EXTRACT ATTACH (LUTYPE6.1) command 86
 EXTRACT ATTACH (MRO) command 88
 resource scheduling 71
 RESP
 deactivating NOHANDLE 132
 option 5
 values in EIBRESP 379
 RESP2
 EXPIRED in messages to console operators 366
 ILLOGIC in SPOOLOPEN ESCAPE 300, 305
 INVREQ in messages to console operators 366
 INVREQ in SIGNOFF command (Security control) 291
 INVREQ in SIGNON (Security control) 293
 INVREQ in SPOOLCLOSE 294
 INVREQ in SPOOLOPEN ESCAPE 300, 305
 INVREQ in WAIT EXTERNAL 353
 INVREQ on WAITCICS 359
 LENGERR in messages to console operators 366
 NOSPOOL in SPOOLCLOSE 294
 NOSPOOL in SPOOLOPEN ESCAPE 300, 305
 NOTAUTH in SIGNON (Security control) 293
 option 5
 SPOOLCLOSE REPORT condition 296
 SPOOLOPEN MAPNAME condition 305
 SPOOLOPEN REPORT (JCL) condition 315
 SPOOLOPEN REPORT condition 314
 SPOOLWRITE MAPNAME condition 327
 SPOOLWRITE REPORT condition 330
 USERIDERR in SIGNON (Security control) 293
 values in EIBRESP2 379
 RESSEC option
 ASSIGN command 25
 RESTART option
 ASSIGN command 25
 RESTYPE option
 QUERY SECURITY command 191
 RESUME option
 SPOOLCLOSE REPORT command 296
 SPOOLOPEN RESUME command 321
 RETAIN option
 SEND PAGE command 279
 RETCODE option
 GDS ALLOCATE command 104
 GDS ASSIGN command 106
 GDS CONNECT PROCESS command 107
 GDS EXTRACT ATTRIBUTES command 109
 GDS EXTRACT PROCESS command 110
 GDS FREE command 111
 GDS ISSUE ABEND command 112
 GDS ISSUE CONFIRMATION command 113
 GDS ISSUE ERROR command 114
 GDS ISSUE PREPARE command 115
 GDS ISSUE SIGNAL command 116
 GDS RECEIVE command 117
 GDS SEND command 119
 GDS WAIT command 121
 RETPAGE condition
 SEND CONTROL command 271
 SEND MAP command 277
 SEND PAGE command 279
 SEND TEXT command 285
 RETRIEVE command 237
 retrieve data stored for task 237
 RETURN command 240
 return program control 240
 RETURNPROG option
 ASSIGN command 25
 REVERSE value
 DFHMDF 417
 DFHMDI 424
 DFHMDS 430
 REWRITE command 242
 REWRITE option
 WRITEQ TS command 370
 RIDFLD option
 DELETE command 66
 ISSUE ADD command 139
 ISSUE ERASE command 151
 ISSUE NOTE command 156
 ISSUE REPLACE command 164
 READ command 193
 READNEXT command 198
 READPREV command 202
 RESETBR command 235
 STARTBR command 338
 WRITE command 361
 RIGHT value
 DFHMDF 417
 DFHMDI 424
 ROLLBACK option
 SYNCPOINT ROLLBACK command 342
 ROLLEDBACK condition
 LINK command 176
 SYNCPOINT command 341

ROUTE command 244
 ROUTECODES option
 WRITE OPERATOR command 366
 RPROCESS option
 BUILD ATTACH (LUTYPE6.1) command 30
 BUILD ATTACH (MRO) command 32
 EXTRACT ATTACH (LUTYPE6.1) command 86
 EXTRACT ATTACH (MRO) command 88
 RRESOURCE option
 BUILD ATTACH (LUTYPE6.1) command 30
 BUILD ATTACH (MRO) command 32
 EXTRACT ATTACH (LUTYPE6.1) command 86
 EXTRACT ATTACH (MRO) command 88
 RRN option
 DELETE command 66
 ISSUE ADD command 139
 ISSUE ERASE command 151
 ISSUE NOTE command 156
 ISSUE REPLACE command 164
 READ command 193
 READNEXT command 199
 READPREV command 202
 RESETBR command 235
 STARTBR command 338
 WRITE command 361
 RSL option
 SPOOLOPEN ESCAPE command 299
 SPOOLOPEN MAPNAME command 304
 SPOOLOPEN REPORT (SCS/T3270) command 319
 SPOOLOPEN REPORT command 313
 RTEFAIL condition
 ROUTE command 246
 RTERMID option
 RETRIEVE command 237
 START command 334
 RTESOME condition
 ROUTE command 246
 RTRANSID option
 RETRIEVE command 238
 START command 334

S

SAA (Systems Application Architecture)
 communications (CPI) 395
 Resource Recovery 393
 SAME value
 DFHMDI 422, 425
 SCHD (DL/I) command 247
 schedule use of resource by task 71, 80
 SCHEDULE, full form of SCHD (DL/I) command 247
 SCRNHNT option
 ASSIGN command 25
 SCRNRWD option
 ASSIGN command 25

SCS (SNA character string)
 CONVERSE command 42
 SEND (VTAM) command 256
 SEND command 250
 SCS option
 SPOOLOPEN REPORT (SCS/T3270) command 319
 SCS printer logical unit, 3790 256
 SCS/T3270 (SNA character string)
 SPOOLOPEN REPORT (SCS/T3270) 317
 SECONDS option
 DELAY command 63
 POST command 186
 ROUTE command 245
 START command 334
 security commands 8
 SEGMENTLIST option
 DUMP TRANSACTION command 76
 SELNERR condition
 ISSUE ABORT command 138
 ISSUE ADD command 140
 ISSUE END command 147
 ISSUE ERASE command 152
 ISSUE NOTE command 156
 ISSUE QUERY command 161
 ISSUE REPLACE command 165
 ISSUE SEND command 168
 ISSUE WAIT command 172
 SEND (2260) command 263
 SEND (2741) command 263
 SEND (2980) command 264
 SEND (3270 display) command 264
 SEND (3270 logical) command 251
 SEND (3600 BTAM) command 265
 SEND (3600 pipeline) command 251
 SEND (3600-3601) command 252
 SEND (3600-3614) command 252
 SEND (3650 interpreter) command 253
 SEND (3650-3270) command 253
 SEND (3650-3653) command 254
 SEND (3650-3680) command 254
 SEND (3735) command 265
 SEND (3740) command 266
 SEND (3767) command 255
 SEND (3770) command 255
 SEND (3790 3270-display) command 257
 SEND (3790 3270-printer) command 257
 SEND (3790 full-function or inquiry) command 256
 SEND (3790 SCS) command 256
 SEND (APPC) command 248
 SEND (LUTYPE2/LUTYPE3) command 249
 SEND (LUTYPE4) command 249
 SEND (LUTYPE6.1) command 250
 SEND (MRO) command 261
 SEND (non-VTAM default) command 261
 SEND (SCS) command 250

- SEND (System/3) command 262
- SEND (System/7) command 262
- SEND (vtam default) command 248
- send asynchronous interrupt 388
- SEND command
 - write to terminal 387
- SEND CONTROL command 269
- SEND MAP command 273
- SEND PAGE command 278
- SEND PARTNSET command 281
- SEND TEXT command 282
- SEND TEXT MAPPED command 286
- SEND TEXT NOEDIT command 288
- sending data to output device 167
- SEP option
 - SPOOLOPEN ESCAPE command 299
 - SPOOLOPEN MAPNAME command 305
 - SPOOLOPEN REPORT (SCS/T3270) command 319
 - SPOOLOPEN REPORT command 313
- sequential retrieval, browsing
 - reading records 192
- SESSBUSY condition
 - ALLOCATE (LUTYPE6.1) command 16
- SESSION option
 - ALLOCATE (LUTYPE6.1) command 16
 - CONNECT PROCESS command 38
 - CONVERSE (non-VTAM) command 61
 - CONVERSE (VTAM) command 50
 - EXTRACT ATTACH (LUTYPE6.1) command 86
 - EXTRACT ATTACH (MRO) command 88
 - EXTRACT ATTRIBUTES (MRO) command 90
 - FREE (LUTYPE6.1) command 100
 - FREE (MRO) command 101
 - ISSUE DISCONNECT (LUTYPE6.1) command 145
 - ISSUE SIGNAL (LUTYPE6.1) command 170
 - POINT command 183
 - RECEIVE (non-VTAM) command 226
 - RECEIVE (VTAM) command 215
 - SEND (non-VTAM) command 267
 - SEND (VTAM) command 259
 - WAIT TERMINAL command 357
- session, passing 157
- SESSIONERR condition
 - ALLOCATE (LUTYPE6.1) command 16
 - EIBRCODE bytes 1-2 377
- SET option
 - ADDRESS SET command 13
 - CONVERSE (non-VTAM) command 61
 - CONVERSE (VTAM) command 50
 - EXTRACT LOGONMSG command 91
 - GDS RECEIVE command 117
 - GETMAIN command 123
 - ISSUE RECEIVE command 162
 - LOAD command 178
 - POST command 186
 - READ command 193
- SET option (*continued*)
 - READNEXT command 199
 - READPREV command 202
 - READQ TD command 204
 - READQ TS command 206
 - RECEIVE (non-VTAM) command 226
 - RECEIVE (VTAM) command 215
 - RECEIVE MAP command 229
 - RECEIVE PARTN command 231
 - RETRIEVE command 238
 - SEND CONTROL command 271
 - SEND MAP command 276
 - SEND PAGE command 279
 - SEND TEXT command 285
- SHARED option
 - GETMAIN command 123
- SIGDATA option
 - ASSIGN command 25
- SIGNAL condition
 - CONVERSE (VTAM) command 51
 - ISSUE CONFIRMATION command 141
 - ISSUE DISCONNECT (default) command 144
 - ISSUE ERROR command 154
 - RECEIVE (VTAM) command 216
 - SEND (VTAM) command 259
 - WAIT SIGNAL command 356
 - WAIT TERMINAL command 357
- SIGNOFF command 291
- SIGNON command 292
- SIT option
 - DUMP TRANSACTION command 76
- SIZE operand
 - DFHMDI 425
- SNA character string
 - See SCS
- SOSI operand
 - DFHMDF 419
 - DFHMDI 425
 - DFHMDS 431
- SOSI option
 - ASSIGN command 25
- SOSI1 option
 - SPOOLOPEN REPORT (SCS/T3270) command 319
 - SPOOLOPEN REPORT command 313
- SOSI2 option
 - SPOOLOPEN REPORT (SCS/T3270) command 319
 - SPOOLOPEN REPORT command 313
- SPCOMMAND
 - RESID value not valid 191
 - RESTYPE values 191
- Spool commands 8
- SPOOLCLOSE command 294
- SPOOLCLOSE REPORT command 295
- SPOOLOPEN ESCAPE command 298
- SPOOLOPEN INPUT command 301

SPOOLOPEN MAPNAME command 303
 SPOOLOPEN OUTPUT command 307
 SPOOLOPEN REPORT (JCL) command 315
 SPOOLOPEN REPORT (SCS/T3270) command 317
 SPOOLOPEN REPORT command 310
 SPOOLOPEN RESUME command 321
 SPOOLREAD command 323
 SPOOLWRITE command 324
 SPOOLWRITE MAPNAME command 326
 SPOOLWRITE REPORT command 329
 standard attribute and printer control character list, BMS (DFHBMSCA) 405
 START ATTACH command 336
 START command 331
 STARTBR command 337
 STARTCODE option
 ASSIGN command 26
 STARTIO option
 WAIT JOURNALNUM command 355
 WRITE JOURNALNUM command 363
 STATE option
 ALLOCATE (APPC) command 14
 ALLOCATE (MRO) command 18
 CONNECT PROCESS command 38
 CONVERSE (non-VTAM) command 61
 CONVERSE (VTAM) command 50
 EXTRACT ATTRIBUTES (APPC) command 89
 EXTRACT ATTRIBUTES (MRO) command 90
 FREE (APPC) command 99
 FREE (MRO) command 101
 GDS ALLOCATE command 104
 GDS CONNECT PROCESS command 107
 GDS EXTRACT ATTRIBUTES command 109
 GDS FREE command 111
 GDS ISSUE ABEND command 112
 GDS ISSUE CONFIRMATION command 113
 GDS ISSUE ERROR command 114
 GDS ISSUE PREPARE command 115
 GDS ISSUE SIGNAL command 116
 GDS RECEIVE command 118
 GDS SEND command 119
 GDS WAIT command 121
 ISSUE ABEND command 136
 ISSUE CONFIRMATION command 141
 ISSUE ERROR command 154
 ISSUE PREPARE command 159
 ISSUE SIGNAL (APPC) command 169
 RECEIVE (non-VTAM) command 226
 RECEIVE (VTAM) command 216
 SEND (non-VTAM) command 267
 SEND (VTAM) command 259
 WAIT CONVID command 349
 STATIONID option
 ASSIGN command 26
 storage area length 20
 storage control commands 8
 STORAGE operand
 DFHMSD 431
 STORAGE option
 DUMP TRANSACTION command 76
 storage, dynamic 385
 STRFIELD option
 CONVERSE (non-VTAM) command 61
 CONVERSE (VTAM) command 50
 SEND (non-VTAM) command 267
 SEND (VTAM) command 259
 stub, program 384
 SUBADDR option
 ISSUE ABORT command 137
 ISSUE END command 146
 ISSUE SEND command 167
 ISSUE WAIT command 171
 SUFFIX operand
 DFHMSD 431
 DFHPSD 434
 SUPPRESSED condition
 DUMP TRANSACTION command 77
 WRITE command 362
 SUSPEND command 340
 switched line disconnection 388
 symbolic register DFHEIPLR 386
 synchronization levels
 basic conversations 110
 synchronize, action
 journal output (WAIT JOURNALNUM) 355
 terminal input/output 387
 SYNCLEVEL option
 CONNECT PROCESS command 39
 EXTRACT PROCESS command 92
 GDS CONNECT PROCESS command 108
 GDS EXTRACT PROCESS command 110
 SYNCONRETURN option
 LINK command 175
 syncpoint
 backing out 342
 commands 8
 establishing 341
 SYNCPOINT command 341
 SYNCPOINT ROLLBACK command 342
 syntax notation 1
 SYSBUSY condition
 ALLOCATE (APPC) command 15
 ALLOCATE (LUTYPE6.1) command 16
 ALLOCATE (MRO) command 18
 EIBRCODE byte 3 377
 SYSID option
 ALLOCATE (APPC) command 14
 ALLOCATE (LUTYPE6.1) command 16
 ALLOCATE (MRO) command 18
 ASSIGN command 26
 CANCEL command 33

SYSID option *(continued)*
 DELETE command 66
 DELETEQ TD command 69
 DELETEQ TS command 70, 370
 ENDBR command 78
 EXTRACT TCT command 94
 GDS ALLOCATE command 105
 LINK command 175
 READ command 194
 READNEXT command 199
 READPREV command 202
 READQ TD command 205
 READQ TS command 207
 RESETBR command 235
 REWRITE command 242
 START command 334
 STARTBR command 338
 UNLOCK command 345
 WRITE command 361
 WRITEQ TD command 367
 SYSIDERR condition
 ALLOCATE (APPC) command 15
 ALLOCATE (LUTYPE6.1) command 17
 ALLOCATE (MRO) command 18
 CANCEL command 34
 DELETE command 67
 DELETEQ TD command 69
 DELETEQ TS command 70
 EIBRCODE bytes 1-2 377
 ENDBR command 79
 LINK command 177
 READ command 195
 READNEXT command 200
 READPREV command 203
 READQ TD command 205
 READQ TS command 207
 RESETBR command 236
 REWRITE command 243
 START command 335
 STARTBR command 339
 UNLOCK command 346
 WRITE command 362
 WRITEQ TD command 368
 WRITEQ TS command 370
 System/3
 CONVERSE command 53
 RECEIVE command 218
 SEND command 262
 System/7
 CONVERSE command 53
 RECEIVE command 218
 SEND command 262
 systemname
 definition 2, 3, 4
 systemname argument, CICS command format 2

T

T3270 option
 SPOOL OPEN REPORT (SCS/T3270) command 319
 TABLES option
 DUMP TRANSACTION command 76
 tables, loading 178
 task
 initiation 333
 task control commands 8
 TASK option
 DUMP TRANSACTION command 76
 task, abnormal termination 128
 task, delay processing of 63
 TASKDATALOC resource definition option 11
 TASKPRIORITY option
 ASSIGN command 26
 TCT option
 DUMP TRANSACTION command 76
 TCTUA option
 ADDRESS command 12
 TCTUALENG option
 ASSIGN command 26
 teletypewriter
 messages 389
 programming 389
 TELLERID option
 ASSIGN command 26
 temporary storage control commands 8
 TERM (DL/I) command 343
 TERM operand
 DFHMSD 431
 TERM CODE option
 ASSIGN command 26, 381
 TERMERR condition
 CONNECT PROCESS command 39
 CONVERSE (non-VTAM) command 62
 CONVERSE (VTAM) command 51
 ISSUE ABEND command 136
 ISSUE CONFIRMATION command 141
 ISSUE COPY (3270 logical) command 143
 ISSUE DISCONNECT (default) command 144
 ISSUE DISCONNECT (LUTYPE6.1) command 145
 ISSUE EODS command 150
 ISSUE ERASEAUP command 153
 ISSUE ERROR command 154
 ISSUE LOAD command 155
 ISSUE PREPARE command 159
 ISSUE PRINT command 160
 ISSUE SIGNAL (APPC) command 169
 ISSUE SIGNAL (LUTYPE6.1) command 170
 LINK command 177
 RECEIVE (non-VTAM) command 226
 RECEIVE (VTAM) command 216
 SEND (non-VTAM) command 267
 SEND (VTAM) command 260

TERMERR condition (*continued*)
 WAIT SIGNAL command 356
 TERMID option
 EXTRACT TCT command 94
 ISSUE COPY (3270 display) command 142
 ISSUE COPY (3270 logical) command 143
 START command 334
 TERMIDERR condition
 ISSUE COPY (3270 display) command 142
 START command 335
 terminal control 387
 commands 8
 terminal model codes 381
 terminal operator paging, initiate paging transaction 278
 TERMINAL option
 DUMP TRANSACTION command 76
 RECEIVE MAP command 229
 SEND CONTROL command 271
 SEND MAP command 276
 SEND TEXT command 285
 SEND TEXT MAPPED command 286
 SEND TEXT NOEDIT command 289
 terminal type codes 381
 terminate data set processing
 abnormal 137
 normal 146
 TERMINATE, full form of TERM (DL/I) command 343
 TERMPRIORITY option
 ASSIGN command 26
 TEXT option
 WRITE OPERATOR command 366
 TEXTKYBD option
 ASSIGN command 26
 TEXTLENGTH option
 WRITE OPERATOR command 366
 TEXTPRINT option
 ASSIGN command 26
 time of day, requesting 19
 TIME option
 DELAY command 63
 FORMATTIME command 96
 POST command 186
 ROUTE command 245
 START command 334
 TIMEOUT option
 WRITE OPERATOR command 366
 timer-event control area 185
 TIMESEP option
 FORMATTIME command 96
 TIOAPFX operand
 DFHMDI 425
 DFHMSD 432
 TITLE option
 ROUTE command 246
 SPOOLOPEN ESCAPE command 300
 SPOOLOPEN MAPNAME command 305
 TITLE option (*continued*)
 SPOOLOPEN REPORT (SCS/T3270) command 319
 SPOOLOPEN REPORT command 313
 TOFLENGTH option
 CONVERSE (non-VTAM) command 61
 CONVERSE (VTAM) command 50
 fullword alternative to TOFLENGTH 387
 SPOOLREAD 323
 TOKEN option
 DELETE command 66
 READ command 194
 REWRITE command 242
 SPOOLCLOSE 294
 SPOOLCLOSE REPORT command 296
 SPOOLOPEN ESCAPE command 300
 SPOOLOPEN INPUT 301
 SPOOLOPEN MAPNAME command 305
 SPOOLOPEN OUTPUT 308
 SPOOLOPEN REPORT (JCL) command 315
 SPOOLOPEN REPORT (SCS/T3270) command 319
 SPOOLOPEN REPORT command 313
 SPOOLOPEN RESUME command 321
 SPOOLREAD 323
 SPOOLWRITE 324
 SPOOLWRITE MAPNAME command 327
 SPOOLWRITE REPORT command 330
 UNLOCK command 345
 TOLENGTH option
 CONVERSE (non-VTAM) command 61
 CONVERSE (VTAM) command 50
 fullword length alternative (TOFLENGTH) 387
 TRACE command 344
 trace control
 controlling trace facility 344
 user trace entry point 82
 TRACENUM option
 ENTER TRACENUM command 83
 TRAILER operand
 DFHMDI 426
 TRAILER option
 SEND PAGE command 279
 SEND TEXT command 285
 TRANPRIORITY option
 ASSIGN command 26
 transfer program control 371
 TRANSID option
 CANCEL command 33
 LINK command 175
 RETURN command 241
 SEND PAGE command 279
 START ATTACH command 336
 START command 334
 TRANSIDERR condition
 START ATTACH command 336
 START command 335

- transient data commands 9
- transient data control
 - delete intrapartition queue 69
 - read data from TD queue 204
 - write data to TD queue 367
- translated code 383
- TRANSP operand
 - DFHMDF 419
 - DFHMDI 426
 - DFHMSD 432
- TRAP(ON), LE/VSE run-time option 10
- TRIGGER option
 - HANDLE AID command 130
- TRIGGER value
 - DFHMDF 419
 - DFHMDI 426
 - DFHMSD 432
- TRIGRAPH operand
 - DFHMSD 432
- TRT option
 - DUMP TRANSACTION command 77
- TSIOERR condition
 - PURGE MESSAGE command 188
 - SEND CONTROL command 272
 - SEND MAP command 277
 - SEND PAGE command 279
 - SEND TEXT command 285
 - SEND TEXT MAPPED command 287
 - SEND TEXT NOEDIT command 290
- TWA option
 - ADDRESS command 12
- TWALENG option
 - ASSIGN command 26
- type codes (terminal) 381
- TYPE operand
 - DFHMDI 426
 - DFHMSD 432

U

- UNATTEND option
 - ASSIGN command 26
- UNDERLINE value
 - DFHMDF 417
 - DFHMDI 424
 - DFHMSD 430
- UNEXPIN condition
 - ISSUE ABORT command 138
 - ISSUE ADD command 140
 - ISSUE END command 147
 - ISSUE ERASE command 152
 - ISSUE NOTE command 156
 - ISSUE QUERY command 161
 - ISSUE RECEIVE command 163
 - ISSUE REPLACE command 165
 - ISSUE SEND command 168

- UNEXPIN condition (*continued*)
 - ISSUE WAIT command 172
 - RECEIVE MAP command 229
- UNLOCK command 345
- UNPROT value
 - DFHMDF 416
- UNTIL option
 - DELAY command 64
- UPDATE option
 - QUERY SECURITY command 191
 - READ command 194
- updating records
 - batch data interchange 164
 - file control 242
- user trace entry point 82
- USERDATA option
 - SPOOLOPEN ESCAPE command 300
 - SPOOLOPEN MAPNAME command 305
 - SPOOLOPEN REPORT (SCS/T3270) command 319
 - SPOOLOPEN REPORT command 313
- USERDATAKEY option
 - GETMAIN command 124
- USERID option
 - ASSIGN command 26
 - CHANGE PASSWORD command 35
 - SIGNON command 293
 - SPOOLOPEN ESCAPE command 300
 - SPOOLOPEN INPUT 301
 - SPOOLOPEN MAPNAME command 305
 - SPOOLOPEN OUTPUT 308
 - SPOOLOPEN REPORT (SCS/T3270) command 320
 - SPOOLOPEN REPORT command 313
 - START command 335
 - VERIFY PASSWORD command 347
- USERIDERR condition
 - CHANGE PASSWORD command 35
 - SIGNON command 293
 - START command 335
 - VERIFY PASSWORD command 348
- USERNAME option
 - ASSIGN command 26
- USERPRIORITY option
 - ASSIGN command 26
- USING option
 - ADDRESS SET command 13

V

- VALIDATION option
 - ASSIGN command 26
- VALIDN operand
 - DFHMDF 419
 - DFHMDI 426
 - DFHMSD 432
- VERIFY PASSWORD command 347

VIEWPOS operand
 DFHPDI 433

VIEWSZE operand
 DFHPDI 433

VOLUME option
 ISSUE ABORT command 137
 ISSUE ADD command 139
 ISSUE END command 146
 ISSUE ERASE command 151
 ISSUE NOTE command 156
 ISSUE QUERY command 161
 ISSUE REPLACE command 165
 ISSUE SEND command 167
 ISSUE WAIT command 171

VOLUMELENG option
 ISSUE ABORT command 137
 ISSUE ADD command 139
 ISSUE END command 146
 ISSUE ERASE command 151
 ISSUE NOTE command 156
 ISSUE QUERY command 161
 ISSUE REPLACE command 165
 ISSUE SEND command 168
 ISSUE WAIT command 171

VSAM WRITE MASSINSERT
 DISABLED cannot occur 345
 NOTOPEN cannot occur 346
 terminate operation 345

VTAB operand
 DFHMSD 432

VTAM logon data, access to 91

W

WAIT CONVID (APPC) command 349

WAIT EVENT command 350

WAIT EXTERNAL command 352

WAIT JOURNAL command 354

WAIT JOURNALNUM command 355

WAIT option
 GDS SEND command 120
 ISSUE COPY (3270 display) command 142
 ISSUE COPY (3270 logical) command 143
 ISSUE ERASEAUP command 153
 RETRIEVE command 238
 SEND (non-VTAM) command 267
 SEND (VTAM) command 259
 SEND command 387
 SEND CONTROL command 271
 SEND MAP command 276
 SEND TEXT command 285
 SEND TEXT MAPPED command 286
 SEND TEXT NOEDIT command 289
 terminal control 387
 WRITE JOURNALNUM command 363

WAIT SIGNAL command 356

WAIT TERMINAL command 357
 general information 387

WAITCICS command 358

waits
 batch data interchange 171
 for event to occur 350
 terminal control operation 387

WPMEDIA option
 ISSUE ABORT command 137
 ISSUE END command 146
 ISSUE SEND command 168
 ISSUE WAIT command 171

WRBRK condition
 CONVERSE (non-VTAM) command 62
 SEND (non-VTAM) command 268
 SEND CONTROL command 272
 SEND MAP command 277
 SEND PAGE command 280
 SEND TEXT command 285
 SEND TEXT MAPPED command 287
 SEND TEXT NOEDIT command 290

write break 55

WRITE command 360

WRITE JOURNALNUM command 363

WRITE OPERATOR command 365
 critical action 365
 eventual action 365
 immediate action 365

write trace entry (ENTER TRACEID) command 82

WRITEQ TD command 367

WRITEQ TS command 369

writing data
 to temporary storage queue 369
 to terminal or logical unit 387
 to transient data queue 367

writing records to data sets
 batch data interchange 139
 file control 360

WRONGSTAT condition
 SPOOLOPEN RESUME command 322

X

XCTL command 371

XINIT operand
 DFHMDF 420

XRF, generic applid 21

Y

YEAR option
 FORMATTIME command 96

YES value
 DFHMDF 423, 425
 DFHMSD 430, 432

YYDDD option	
FORMATTIME command	96
YYDDMM option	
FORMATTIME command	96
YYMMDD option	
FORMATTIME command	96
YYYYDDD option	
FORMATTIME command	96
YYYYDDMM option	
FORMATTIME command	96
YYYYMMDD option	
FORMATTIME command	97

Z

ZERO value	
DFHMDF	417

Sending your comments to IBM

CICS® Transaction Server for VSE/ESA™

Application Programming Reference

SC33-1658-04

If you want to send to IBM any comments you have about this book, please use one of the methods listed below. Feel free to comment on anything you regard as a specific error or omission in the subject matter, and on the clarity, organization or completeness of the book itself.

To request additional publications, or to ask questions or make comments about the functions of IBM products or systems, you should talk to your IBM representative or to your IBM authorized remarketer.

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate, without incurring any obligation to you.

You can send your comments to IBM in any of the following ways:

- By mail:
 - User Technologies Department (MP 095)
 - IBM United Kingdom Laboratories
 - Hursley Park
 - WINCHESTER
 - Hampshire
 - SO21 2JN
 - United Kingdom
- By fax:
 - From outside the U.K., after your international access code use 44 1962 842327
 - From within the U.K., use 01962 842327
- Electronically, use the appropriate network ID:
 - IBM Mail Exchange: GBIBM2Q9 at IBMMAIL
 - IBMLink: HURSLEY(IDRCF)
 - Email: idrcf@hursley.ibm.com

Whichever method you use, ensure that you include:

- The publication number and title
- The page number or topic to which your comment applies
- Your name and address/telephone number/fax number/network ID.



Program Number: 5648-054

SC33-1658-04

