

CICS® Transaction Server for OS/390®



CICS Application Programming Reference

Release 3

CICS® Transaction Server for OS/390®



CICS Application Programming Reference

Release 3

Note!

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

Fourth edition (November 2000)

This edition applies to Release 3 of CICS Transaction Server for OS/390, program number 5655-147, 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.

This edition replaces and makes obsolete the previous edition, SC33-1688-02. Changes since that edition are indicated by a '#' sign to the left of a change. Any vertical lines in the left margin indicated a change made between Version 1 Release 2 and Version 1 Release 3 of CICS Transaction Server for OS/390.

Order publications through your IBM representative or the IBM branch office serving your locality. Publications are not stocked at the address given below.

At the back of this publication is a page entitled “Sending your comments to IBM”. If you want to make comments, but the methods described are not available to you, please address them to:

IBM United Kingdom Laboratories, Information Development,
Mail Point 095, Hursley Park, Winchester, Hampshire, England, SO21 2JN.

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

© Copyright International Business Machines Corporation 1989, 2000. All rights reserved.

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

Contents

Notices	ix
Programming interface information	x
Trademarks	x
Preface	xi
What this book is about	xi
Who this book is for	xi
What you need to know to understand this book	xi
How to use this book	xi
What this book does not cover	xi
Bibliography	xiii
CICS Transaction Server for OS/390	xiii
CICS books for CICS Transaction Server for OS/390	xiii
CICSplex SM books for CICS Transaction Server for OS/390	xiv
Other CICS books	xiv
Books from related libraries	xiv
MVS	xiv
IMS	xiv
Systems Network Architecture	xv
Systems Application Architecture	xv
SQL	xv
Other related books	xv
Determining if a publication is current	xv
Summary of changes	xvii
Changes for the CICS Transaction Server for OS/390 for release 3 third edition	xvii
CICS Transaction Server for OS/390 for release 2	xvii
CICS Transaction Server for OS/390 for release 1	xvii
Chapter 1. CICS API commands	1
Command format	1
CICS syntax notation used in this book	1
Argument values	2
Restrictions	5
CICS-value data areas (cvdas)	5
LENGTH options	5
NOHANDLE option	5
RESP and RESP2 options	5
Possible ABEND AICA timeout	6
CICS interface to JES	6
Using the CICS interface to JES	8
Commands by function	9
ABEND	13
ADDRESS	14
ADDRESS SET	15
ALLOCATE (APPC)	16
ALLOCATE (LUTYPE6.1)	18
ALLOCATE (MRO)	20
ASKTIME	21
ASSIGN	22
BIF DEEDIT	30
BUILD ATTACH (LUTYPE6.1)	31
BUILD ATTACH (MRO)	33
CANCEL	35
CHANGE PASSWORD	37

CHANGE TASK	38
CONNECT PROCESS	39
CONVERSE (VTAM default)	41
CONVERSE (APPC)	41
CONVERSE (LUTYPE2/LUTYPE3)	42
CONVERSE (LUTYPE4)	42
CONVERSE (LUTYPE6.1)	43
CONVERSE (SCS)	43
CONVERSE (3270 logical)	44
CONVERSE (3600-3601)	44
CONVERSE (3600-3614)	45
CONVERSE (3650 interpreter)	45
CONVERSE (3650-3270)	46
CONVERSE (3650-3653)	46
CONVERSE (3650-3680)	47
CONVERSE (3767)	47
CONVERSE (3770)	48
CONVERSE (3790 full-function or inquiry)	48
CONVERSE (3790 3270-display)	49
CONVERSE (non-VTAM default)	53
CONVERSE (MRO)	53
CONVERSE (System/3)	54
CONVERSE (System/7)	54
CONVERSE (2260)	55
CONVERSE (2741)	56
CONVERSE (2770)	57
CONVERSE (2780)	57
CONVERSE (3270 display)	58
CONVERSE (3600 BTAM)	58
CONVERSE (3735)	60
CONVERSE (3740)	60
DEFINE COUNTER	64
DELAY	66
DELETE	68
DELETE COUNTER	73
DELETEQ TD	75
DELETEQ TS	76
DEQ	77
DOCUMENT CREATE	79
DOCUMENT INSERT	82
DOCUMENT RETRIEVE	84
DOCUMENT SET	85
DUMP TRANSACTION	87
ENDBR	90
ENQ	92
ENTER TRACENUM	94
EXTRACT ATTACH (LUTYPE6.1)	96
EXTRACT ATTACH (MRO)	98
EXTRACT ATTRIBUTES (APPC)	100
EXTRACT ATTRIBUTES (MRO)	101
EXTRACT CERTIFICATE	102
EXTRACT LOGONMSG	104
EXTRACT PROCESS	105
EXTRACT TCPIP	107
EXTRACT TCT	109
FORMATTIME	110
FREE	112
FREE (APPC)	113
FREE (LUTYPE6.1)	114
FREE (MRO)	115

FREEMAIN	116
GDS ALLOCATE	118
GDS ASSIGN	120
GDS CONNECT PROCESS	121
GDS EXTRACT ATTRIBUTES	123
GDS EXTRACT PROCESS	124
GDS FREE	125
GDS ISSUE ABEND	126
GDS ISSUE CONFIRMATION	127
GDS ISSUE ERROR	128
GDS ISSUE PREPARE	129
GDS ISSUE SIGNAL	130
GDS RECEIVE	131
GDS SEND	133
GDS WAIT	135
GET COUNTER	136
GETMAIN	139
HANDLE ABEND	141
HANDLE AID	142
HANDLE CONDITION	143
IGNORE CONDITION	144
ISSUE ABEND	145
ISSUE ABORT	146
ISSUE ADD	148
ISSUE CONFIRMATION	150
ISSUE COPY (3270 display)	151
ISSUE COPY (3270 logical)	152
ISSUE DISCONNECT (default)	153
ISSUE DISCONNECT (LUTYPE6.1)	154
ISSUE END	155
ISSUE ENDFILE	156
ISSUE ENDOUTPUT	157
ISSUE EODS	157
ISSUE ERASE	158
ISSUE ERASEAUP	159
ISSUE ERROR	161
ISSUE LOAD	162
ISSUE NOTE	163
ISSUE PASS	164
ISSUE PREPARE	165
ISSUE PRINT	166
ISSUE QUERY	167
ISSUE RECEIVE	168
ISSUE REPLACE	170
ISSUE RESET	171
ISSUE SEND	172
ISSUE SIGNAL (APPC)	174
ISSUE SIGNAL (LUTYPE6.1)	175
ISSUE WAIT	176
JOURNAL	177
LINK	178
LOAD	183
MONITOR	186
POINT	188
POP HANDLE	188
POST	189
PURGE MESSAGE	191
PUSH HANDLE	192
QUERY COUNTER	193
QUERY SECURITY	195

READ	197
READNEXT	204
READPREV	211
READQ TD	217
READQ TS	219
RECEIVE (VTAM default)	221
RECEIVE (APPC)	221
RECEIVE (LUTYPE2/LUTYPE3)	222
RECEIVE (LUTYPE4)	222
RECEIVE (LUTYPE6.1)	223
RECEIVE (3270 logical)	223
RECEIVE (3600 pipeline)	224
RECEIVE (3600-3601)	224
RECEIVE (3600-3614)	225
RECEIVE (3650)	225
RECEIVE (3767)	226
RECEIVE (3770)	226
RECEIVE (3790 full-function or inquiry)	227
RECEIVE (non-VTAM default)	230
RECEIVE (MRO)	230
RECEIVE (System/3)	231
RECEIVE (System/7)	231
RECEIVE (2260)	232
RECEIVE (2741)	232
RECEIVE (2980)	233
RECEIVE (3270 display)	235
RECEIVE (3600 BTAM)	235
RECEIVE (3735)	236
RECEIVE (3740)	236
RECEIVE (3790 3270-display)	237
RECEIVE MAP	240
RECEIVE MAP MAPPINGDEV	242
RECEIVE PARTN	244
RELEASE	245
RESETBR	247
RETRIEVE	249
RETURN	252
REWIND COUNTER	254
REWRITE	256
ROUTE	259
SEND (VTAM default)	262
SEND (APPC)	262
SEND (LUTYPE2/LUTYPE3)	263
SEND (LUTYPE4)	263
SEND (LUTYPE6.1)	264
SEND (SCS)	264
SEND (3270 logical)	265
SEND (3600 pipeline)	265
SEND (3600-3601)	266
SEND (3600-3614)	266
SEND (3650 interpreter)	267
SEND (3650-3270)	267
SEND (3650-3653)	268
SEND (3650-3680)	268
SEND (3767)	269
SEND (3770)	269
SEND (3790 full-function or inquiry)	270
SEND (3790 SCS)	270
SEND (3790 3270-display)	271
SEND (3790 3270-printer)	271

	SEND (non-VTAM default)	275
	SEND (MRO)	275
	SEND (System/3)	276
	SEND (System/7)	276
	SEND (2260)	277
	SEND (2741)	277
	SEND (2980)	278
	SEND (3270 display)	278
	SEND (3600 BTAM)	279
	SEND (3735)	279
	SEND (3740)	280
	SEND CONTROL	283
	SEND MAP	286
	SEND MAP MAPPINGDEV	291
	SEND PAGE	293
	SEND PARTNSET	295
	SEND TEXT	296
	SEND TEXT MAPPED	300
	SEND TEXT NOEDIT	302
	SIGNOFF	304
	SIGNON	305
	SPOOLCLOSE	308
	SPOOLOPEN INPUT	310
	SPOOLOPEN OUTPUT	312
	SPOOLREAD	315
	SPOOLWRITE	317
	START	319
	START ATTACH	324
	START BREXIT	325
	STARTBR	327
	SUSPEND	330
	SYNCPOINT	331
	SYNCPOINT ROLLBACK	331
	UNLOCK	332
	UPDATE COUNTER	334
	VERIFY PASSWORD	336
	WAIT CONVID (APPC)	337
	WAIT EVENT	339
	WAIT EXTERNAL	340
	WAIT JOURNALNAME	342
	WAIT JOURNALNUM	343
	WAIT SIGNAL	344
	WAIT TERMINAL	344
	WAITCICS	345
#	WEB ENDBROWSE FORMFIELD	346
	WEB ENDBROWSE HTTPHEADER	347
	WEB EXTRACT	348
#	WEB READ FORMFIELD	349
	WEB READ HTTPHEADER	350
#	WEB READNEXT FORMFIELD	351
	WEB READNEXT HTTPHEADER	352
	WEB RECEIVE	353
	WEB RETRIEVE	354
	WEB SEND	355
#	WEB STARTBROWSE FORMFIELD	356
	WEB STARTBROWSE HTTPHEADER	357
	WEB WRITE	358
	WRITE	359
	WRITE JOURNALNAME	364
	WRITE JOURNALNUM	366

WRITE OPERATOR	367
WRITEQ TD	369
WRITEQ TS	371
XCTL	374
Appendix A. EXEC interface block	377
EIB fields	377
Appendix B. Codes returned by ASSIGN	387
ASSIGN TERMCODE	387
ASSIGN FCI	387
Appendix C. Translated code for CICS commands	389
COBOL	389
C	389
PL/I	389
Assembler language	390
Appendix D. Terminal control	393
Commands and options for terminals and logical units	393
TCAM-supported terminals and logical units	394
BTAM programmable terminals	394
Teletypewriter programming	395
Display device operations	396
Appendix E. SAA Resource Recovery	399
SRRCMT	399
SRRBACK	399
Appendix F. Common Programming Interface Communications (CPI Communications)	401
CPI Communications language interfaces	401
Appendix G. API restrictions for distributed program link	403
Summary of the restricted API commands	403
List of API commands	403
Appendix H. CVDA numeric values	407
Appendix I. National language codes	409
Appendix J. BMS-related constants	411
Magnetic slot reader (MSR) control value constants, DFHMSRCA	413
MSR control byte values	413
Attention identifier constants, DFHAID	415
Appendix K. BMS macro summary	417
Mapset, map, and field definition	417
Partition set definition	418
DFHMDF	420
DFHMDI	427
DFHMSD	433
DFHPDI	440
DFHPSD	441
Index	443
Sending your comments to IBM	445

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.

Programming interface information

This book is intended to help you write application programs using EXEC CICS commands that obtain the services of CICS.

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

Trademarks

The following terms are trademarks of International Business Machines Corporation in the United States, or other countries, or both:

AFP	DB2	MVS/ESA
BookManager	DFSMS	RACF
CICS	DFSMS/MVS	System/370
CICS/ESA	IBM	VTAM
DATABASE 2	IMS	

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries.

Other company, product, and service names may be trademarks or service marks of others.

Preface

What this book is about

This book describes the CICS Transaction Server for OS/390 EXEC application programming interface. It contains **reference** information needed to prepare COBOL, C, PL/I, and assembler-language application programs, using EXEC 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 Primer (VS COBOL II)* and 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.

What this book does not cover

The EXEC CICS commands for system programming; that is COLLECT, CREATE, DISABLE, ENABLE, INQUIRE, PERFORM, RESYNC, and SET are not covered in this book. You will find them in the *CICS System Programming Reference*.

The EXEC CICS FEPI commands available for use with the CICS Front End Programming Interface feature are not discussed in this book, but in the *CICS/ESA Front End Programming Interface User's Guide*.

| The EXEC CICS commands available for use with CICS business transaction services (BTS)
| are not discussed in this book. You will find a full description in the *CICS Business*
| *Transaction Services* manual.

| The CICS C++ OO programming interface is not described in this book. It is defined in the
| *CICS C++ OO Class Libraries* manual.

| The CICS Java® programming interface is not described here, it is defined in Javadoc HTML
| provided in **dfjcics_docs.zip** in the OS/390 UNIX System Services HFS, in the
| /usr/lpp/cicsts/cicsts13 directory.

Bibliography

CICS Transaction Server for OS/390

<i>CICS Transaction Server for OS/390: Planning for Installation</i>	GC33-1789
<i>CICS Transaction Server for OS/390 Release Guide</i>	GC34-5352
<i>CICS Transaction Server for OS/390 Migration Guide</i>	GC34-5353
<i>CICS Transaction Server for OS/390 Installation Guide</i>	GC33-1681
<i>CICS Transaction Server for OS/390 Program Directory</i>	GI10-2506
<i>CICS Transaction Server for OS/390 Licensed Program Specification</i>	GC33-1707

CICS books for CICS Transaction Server for OS/390

General

<i>CICS Master Index</i>	SC33-1704
<i>CICS User's Handbook</i>	SX33-6104
<i>CICS Transaction Server for OS/390 Glossary (softcopy only)</i>	GC33-1705

Administration

<i>CICS System Definition Guide</i>	SC33-1682
<i>CICS Customization Guide</i>	SC33-1683
<i>CICS Resource Definition Guide</i>	SC33-1684
<i>CICS Operations and Utilities Guide</i>	SC33-1685
<i>CICS Supplied Transactions</i>	SC33-1686

Programming

<i>CICS Application Programming Guide</i>	SC33-1687
<i>CICS Application Programming Reference</i>	SC33-1688
<i>CICS System Programming Reference</i>	SC33-1689
<i>CICS Front End Programming Interface User's Guide</i>	SC33-1692
<i>CICS C++ OO Class Libraries</i>	SC34-5455
<i>CICS Distributed Transaction Programming Guide</i>	SC33-1691
<i>CICS Business Transaction Services</i>	SC34-5268

Diagnosis

<i>CICS Problem Determination Guide</i>	GC33-1693
<i>CICS Messages and Codes</i>	GC33-1694
<i>CICS Diagnosis Reference</i>	LY33-6088
<i>CICS Data Areas</i>	LY33-6089
<i>CICS Trace Entries</i>	SC34-5446
<i>CICS Supplementary Data Areas</i>	LY33-6090

Communication

<i>CICS Intercommunication Guide</i>	SC33-1695
<i>CICS Family: Interproduct Communication</i>	SC33-0824
<i>CICS Family: Communicating from CICS on System/390</i>	SC33-1697
<i>CICS External Interfaces Guide</i>	SC33-1944
<i>CICS Internet Guide</i>	SC34-5445

Special topics

<i>CICS Recovery and Restart Guide</i>	SC33-1698
<i>CICS Performance Guide</i>	SC33-1699
<i>CICS IMS Database Control Guide</i>	SC33-1700
<i>CICS RACF Security Guide</i>	SC33-1701
<i>CICS Shared Data Tables Guide</i>	SC33-1702
<i>CICS Transaction Affinities Utility Guide</i>	SC33-1777
<i>CICS DB2 Guide</i>	SC33-1939

CICSplex SM books for CICS Transaction Server for OS/390

General

<i>CICSplex SM Master Index</i>	SC33-1812
<i>CICSplex SM Concepts and Planning</i>	GC33-0786
<i>CICSplex SM User Interface Guide</i>	SC33-0788
<i>CICSplex SM Web User Interface Guide</i>	SC34-5403
<i>CICSplex SM View Commands Reference Summary</i>	SX33-6099

Administration and Management

<i>CICSplex SM Administration</i>	SC34-5401
<i>CICSplex SM Operations Views Reference</i>	SC33-0789
<i>CICSplex SM Monitor Views Reference</i>	SC34-5402
<i>CICSplex SM Managing Workloads</i>	SC33-1807
<i>CICSplex SM Managing Resource Usage</i>	SC33-1808
<i>CICSplex SM Managing Business Applications</i>	SC33-1809

Programming

<i>CICSplex SM Application Programming Guide</i>	SC34-5457
<i>CICSplex SM Application Programming Reference</i>	SC34-5458

Diagnosis

<i>CICSplex SM Resource Tables Reference</i>	SC33-1220
<i>CICSplex SM Messages and Codes</i>	GC33-0790
<i>CICSplex SM Problem Determination</i>	GC33-0791

Other CICS books

<i>CICS Application Programming Primer (VS COBOL II)</i>	SC33-0674
<i>CICS Application Migration Aid Guide</i>	SC33-0768
<i>CICS Family: API Structure</i>	SC33-1007
<i>CICS Family: Client/Server Programming</i>	SC33-1435
<i>CICS Family: General Information</i>	GC33-0155
<i>CICS 4.1 Sample Applications Guide</i>	SC33-1173
<i>CICS/ESA 3.3 XRF Guide</i>	SC33-0661

If you have any questions about the CICS Transaction Server for OS/390 library, see *CICS Transaction Server for OS/390: Planning for Installation* which discusses both hardcopy and softcopy books and the ways that the books can be ordered.

Books from related libraries

MVS

See the following books:

<i>OS/390 MVS Initialization and Tuning Guide</i>
<i>OS/390 MVS Initialization and Tuning Reference</i>
<i>OS/390 MVS JCL User's Guide</i>
<i>OS/390 MVS System Commands</i>

IMS

For information about IMS, see the following books. If you use the CICS-DL/I interface, see one of the following IMS books:

<i>IMS/VS Version 2 Application Programming for CICS/OS/VS Users</i> , SC26-4177
<i>IMS Application Programming: Design Guide</i> , SC26-4279
<i>IMS Application Programming: EXEC DLI Commands manual</i> , SC26-4280
<i>IMS Application Programming: DL/I Calls manual</i> , SC26-4274.

These books contain information about IMS application programming for CICS users, and tell you how to write online or batch programs that use the EXEC DLI interface or the DL/I CALL interface with the DL/I interface block (DIB).

Note: The EXEC DLI interface is no longer documented in the *CICS Application Programming Reference*.

Systems Network Architecture

See the following Systems Network Architecture (SNA) book for further information about SNA:

Sessions between Logical Units, GC20-1868.

Systems Application Architecture

See the following Systems Application Architecture (SAA) for further information about SAA:

SAA Common Programming Interface Communications Reference, SC26-4399.

SQL

For information about executing SQL in a CICS application program see the following books:

IBM DATABASE 2 Application Programming Guide for CICS Users, SC26-4080

IBM DATABASE 2 Reference, SC26-4078.

Other related books

You may also want to refer to the following IBM books:

Distributed Processing Programming Executive/Distributed Presentation Services (DPPX/DPS):

DPPX/Distributed Presentation Services Version 2 System Programming Guide, SC33-0117

OS/VS COBOL Compiler and Library Programmer's Guide, SC28-6483

VS COBOL II Application Programming Guide, SC26-4045

OS PL/I Version 2 Programming Guide, SC26-4307

An Introduction to the IBM 3270 Information Display System, GA27-2739

3274 Control Unit Reference Summary, GX20-1878.

Component Description: IBM 2721 Portable Audio Terminal, GA27-3029.

IBM 2780 Data Transmission Terminal Component Description, GA27-3035

CICS/ESE 3.3 IBM 3270 Data Stream Device Guide, SC33-0232

IBM 3270 Data Stream Programmer's Reference, GA23-0059

IBM 4700/3600/3630 Guide, SC33-0233

Determining if a publication is current

IBM regularly updates its publications with new and changed information. When first published, both hardcopy and BookManager softcopy versions of a publication are usually in step. However, due to the time required to print and distribute hardcopy books, the BookManager version is more likely to have had last-minute changes made to it before publication.

Subsequent updates will probably be available in softcopy before they are available in hardcopy. This means that at any time from the availability of a release, softcopy versions should be regarded as the most up-to-date.

For CICS Transaction Server books, these softcopy updates appear regularly on the *Transaction Processing and Data Collection Kit* CD-ROM, SK2T-0730-xx. Each reissue of the collection kit is indicated by an updated order number suffix (the -xx part). For example, collection kit SK2T-0730-06 is more up-to-date than SK2T-0730-05. The collection kit is also clearly dated on the cover.

Updates to the softcopy are clearly marked by revision codes (usually a “#” character) to the left of the changes.

Summary of changes

Changes for this fourth edition are marked by the '#' sign to the left of the text.

Changes for the CICS Transaction Server for OS/390 for release 3 third edition

Changes for this fourth edition are marked by a vertical bar to the left of the text.

This book includes information about the new or enhanced facilities introduced by the CICS element of CICS Transaction Server for OS/390 for release 3. Technical changes are shown by vertical bars in the left hand margin.

A new option, BREXIT, has been added to the START command to define a 3270 Bridge exit. To simplify use of the START command, it is now documented as 3 variations: START ATTACH, START BREXIT and the original Interval Control START. This change shows the limited number of options that you can use with BREXIT and ATTACH.

New DOCUMENT and WEB commands have been added for use with CICS Web support.

EXTRACT TCPIP has been added to support the CICS TCP/IP listener.

EXTRACT CERTIFICATE has been added to support the Secure Sockets Layer (SSL).

If the name specified in the RESOURCE option of an ENQ or DEQ command matches the name of an installed ENQMODEL resource, the ENQSCOPE attribute of the resource definition controls the scope of the ENQ or DEQ command, that is, whether it has local or sysplex scope. The syntax of the ENQ or DEQ command is not changed.

Changes have been made to the PICIN and PICOUT options of the BMS macro DFHMDF to support multi-character currency symbols, such as EUR.

CICS Transaction Server for OS/390 for release 2

This book includes information about the new or enhanced facilities introduced by the CICS element of CICS Transaction Server for OS/390 for release 2. Technical changes are shown by vertical bars in the left hand margin.

A new keyword, BRIDGE, has been added to ASSIGN to support the 3270 bridge environment.

The BMS map definition macros are enhanced to build an Application Data Structure (ADS) descriptor record in the mapset. See Appendix K, "BMS macro summary" on page 417.

RECEIVE MAP MAPPINGDEV and SEND MAP MAPPINGDEV have been added to allow mapping of 3270 datastreams without a terminal input/output operation.

CICS Transaction Server for OS/390 for release 1

There are new commands, WAIT JOURNALNAME and WRITE JOURNALNAME, that supersede WAIT JOURNALNUM and WRITE JOURNALNUM.

journalnum(01) no longer signifies the system log but journalname(dfhlog) should be used for this purpose.

New RLS keywords CONSISTENT, REPEATABLE, and UNCOMMITTED, have been added to the read commands.

New options UPDATE and TOKEN have been added to READNEXT and READPREV.

New conditions LOCKED and RECORDBUSY added to file control commands.

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 system programming interface (SPI) are primarily for the use of the system programmer; they are not described in this book. For details of these 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.”

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 must code. You may code more than one of them, in any sequence.
	A set of alternatives—one of which you may code.
	A set of alternatives — any number (including none) of which you may code once, in any sequence.
	Alternatives where A is the default.
	Use with the named section in place of its name. 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

Data-areas and Data-values

Data-areas and data-values are the basic argument types.

The difference between them is the direction in which information flows when a task executes a command. A **data-value** is always, and *exclusively* a sender; it conveys data to CICS that CICS uses to process the command. A **data-area** is a receiver; CICS uses it to return information to the caller. Note that a data-area can also be a sender, for example when the data to be conveyed to CICS is variable length (as in FROM), or where a field is used both for input and output.

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
 - Doubleword *unsigned* binary — PIC 9(18) 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
 - Doubleword *unsigned* binary — PIC 9(18) 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 the name of any base locator for linkage (BLL) cell, or by any COBOL data

name that contains a copy of such a pointer in a BLL cell.

- “ptr-ref” can be replaced by the name of any BLL cell. In VS COBOL II, you can replace “ptr-ref” with a 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 1–8 characters from A–Z, 0–9, \$, @, and #.

“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 #.

- “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.

In OS/VS COBOL, you must code the LENGTH options. In VS COBOL II, 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
 - Doubleword binary — char[8]
 - 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
 - Doubleword binary — char[8]

- 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 1–8 characters from A–Z, 0–9, \$, @, and #.

“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 #.

- “label” is not supported in the C language.
- “hhmmss” can be replaced by an integer constant; otherwise the application is responsible for ensuring that the value passed to CICS is in packed decimal format. The language does not provide a packed decimal type.

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 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)
 - Doubleword binary — CHAR (8)

- 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)
 - Doubleword binary — CHAR (8)
 - 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 1–8 characters from A–Z, 0–9, \$, @, and #.

“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 #.

- “label” can be replaced by any PL/I expression whose value is a label.
- “hhmmss” 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 0HHMMSS+ 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 language reference to a register.

- “name” can be replaced **either** by a character string in single quotation marks, **or** by an assembler-language 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 1–8 characters from A–Z, 0–9, \$, @, and #.

“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 #.

- “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)
```

- “hmmss” 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)
```

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 VS COBOL II, 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 OS/VS COBOL and 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 total length of a journal record is 62KB, which includes the user data, the prefix data, and the 2-byte JTYPEID. You specify this value on the FLENGTH option.

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)
      NOSUSPEND
      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.

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*.

CICS interface to JES

The CICS interface to JES (the Job Entry Subsystem component) provides specialist programmer EXEC commands for accessing the system spool files maintained by JES2 and JES3. You can support the requirements of other products to exchange files with other systems connected through a JES remote spooling communications subsystem (RSCS) network. The term JES is used to refer to both JES2 and JES3.

The CICS interface to JES enables you to:

- Retrieve data for a specific user from the local JES spool. See Figure 1 on page 7.
- Create a file and write records directly to the local JES spool. See Figure 2 on page 7.
- Send a JES spool file to a specific remote destination. See Figure 3 on page 8.

There are internal limits in JES2 and JES3 that you should consider when you are designing applications to use this interface. Some of these internal limits can depend on which release of JES you are using. You should therefore read the following in conjunction with the appropriate JES documentation.

JES2

- Number of SYSOUT data sets
There is an upper limit to the number of data sets that can be created by a single job. If this limit is exceeded during a CICS run, subsequent SPOOLOPEN OUTPUT requests return the ‘ALLOCERR’ condition.
- Output queue and job queue sizes
The number of Job Output Elements and Job Queue Elements may need to be increased to accommodate the additional output processing. Timely processing of the data sets created using this interface minimizes this requirement.
- Spool space
Although the spool space for a data set created using this interface is reused after it has been processed, some control information is retained for the life of the job. You may have to increase the spool file allocation to allow for this.

JES3

- Job queue size
The number of Job Queue Elements may need to be increased to accommodate the additional output processing. Also, more JSAM buffers may be required. Timely processing of the data sets created using this interface minimizes this requirement.
- Spool space
Although the spool space for a data set created using this interface is reused after it has been processed, some control information is retained for the life of the job. You may have to increase the spool file allocation to allow for this.

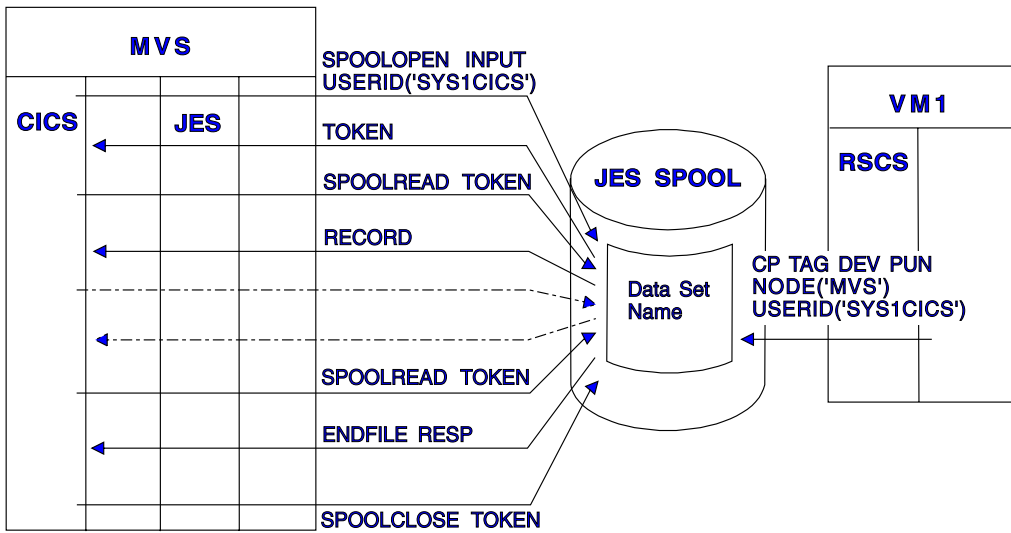


Figure 1. Retrieve data from the JES spool

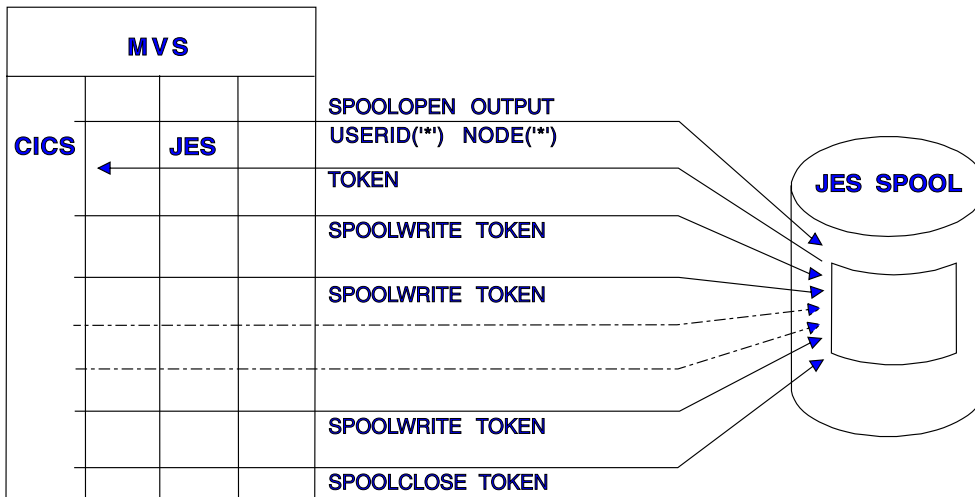


Figure 2. Create a file and write directly to the JES spool

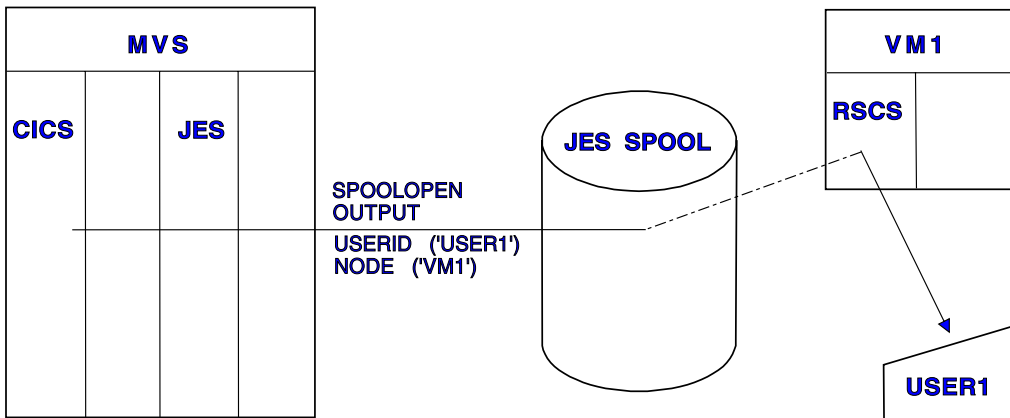


Figure 3. Send the written file to a remote destination

For both JES2 and JES3, some performance degradation can be experienced if a backlog of CICS-created data sets is allowed to accumulate. You should ensure that procedures exist to detect and remedy such situations.

Using the CICS interface to JES

To use the CICS interface to JES, you must code DFHSIT SPOOL=YES.

You must specify RESP or NOHANDLE on the EXEC CICS SPOOLCLOSE, SPOOLOPEN, SPOOLREAD, and SPOOLWRITE commands. RESP bears a one-to-one correspondence with HANDLE CONDITION. If you do not code RESP, your program abends. You can also code the RESP2 option.

NOHANDLE, RESP, and RESP2 are not shown in the syntax boxes or in the option lists for the EXEC CICS SPOOL commands described in this manual. At the end of the description of each spool command, there is a list of RESP values and the RESP2 values that are specific to the CICS-JES interface.

Transactions that process SYSOUT data sets larger than 1000 records, either for INPUT or for OUTPUT, are likely to have a performance impact on the rest of CICS. When you cannot avoid such a transaction, you should carefully evaluate general system performance. You should introduce a pacing mechanism if the effects on the rest of CICS are unacceptable.

All access to a JES spool file must be completed within one logical unit of work. Issuing an EXEC CICS SYNCPOINT command implicitly issues a SPOOLCLOSE command for any open report.

Input

A remote application must route any files intended for a CICS transaction to a specific user name at the system where CICS resides. See Figure 1 on page 7 for an example of a CP command used by a VM system to do this. The figure also shows the EXEC CICS SPOOL commands you use to retrieve the data.

The CICS transaction issues the SPOOLOPEN command, specifying the writer name on the USERID parameter and optionally the class of output within the writer name. The normal response is:

1. No input for this external writer.
2. The single-thread is busy.
3. The file is allocated to you for retrieval, and is identified by the "token" returned by CICS. The token must be included on every SPOOL command for retrieving the data set.

In cases (1) and (2), the transaction should retry the SPOOLOPEN after a suitable interval, by restarting itself.

In case (3), the transaction should then retrieve the file with SPOOLREAD commands, and proceed to SPOOLCLOSE as rapidly as possible to release the path for other users. This is especially important for **input** from JES because the input path is **single-threaded**. When there is more than one transaction using the interface, their files can be differentiated by using different writer names or different classes within a single writer name. Furthermore, you should ensure that the transactions either terminate or wait for a short period between SPOOLCLOSE and a subsequent SPOOLOPEN. If you do not do this, one transaction can prevent others from using the interface.

JES exits

Both JES2 and JES3 provide a way of screening incoming files. For JES2, the TSO/E Interactive Data Transmission Facility Screening and Notification exit is used. The JES3 equivalent is the Validate Incoming Netdata File exit.

You should review any use your installation makes of these exits to ensure that files that are to be read using the CICS interface to JES are correctly processed.

Output

The transaction program issues SPOOLOPEN to allocate an output data set, specifying a remote NODE and USERID. SPOOLOPEN returns a unique token to the transaction, which must be used in all subsequent SPOOLWRITE and SPOOLCLOSE commands to identify the file being written to. Finally, the transaction issues SPOOLCLOSE to close and deallocate the report, and permit its immediate printing or onward routing by the system spooler. The normal response received from the SPOOLOPEN OUTPUT command is:

The file is allocated to you and identified by the token returned by CICS.
Data can now be written to it.

If the node is a remote system, the data set is queued on the JES spool against the destination userid. The ID of this destination user was specified on the SPOOLOPEN OUTPUT USERID parameter. If the node is a remote VM system, the data is queued in the VM RDR queue for the ID that was specified on the same USERID parameter.

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 MAP MAPPINGDEV
RECEIVE PARTN
ROUTE
SEND CONTROL
SEND MAP
| SEND MAP MAPPINGDEV
SEND PAGE
SEND PARTNSET
SEND TEXT
SEND TEXT MAPPED
SEND TEXT NOEDIT

Built-in functions

BIF DEEDIT

Console support

WRITE OPERATOR

Diagnostic services

DUMP TRANSACTION
ENTER TRACENUM

Document services

| DOCUMENT CREATE
| DOCUMENT INSERT
| DOCUMENT RETRIEVE
| DOCUMENT SET

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

WAIT JOURNALNAME
WAIT JOURNALNUM
WRITE JOURNALNAME
WRITE JOURNALNUM

Monitoring

MONITOR

Named counter server

| DEFINE COUNTER
| DEFINE DCOUNTER
| DELETE COUNTER
| DELETE DCOUNTER
| GET COUNTER
| GET DCOUNTER
| QUERY COUNTER
| QUERY DCOUNTER
| REWIND COUNTER
| REWIND DCOUNTER
| UPDATE COUNTER
| UPDATE DCOUNTER

Program control

LINK
LOAD
RELEASE
RETURN
XCTL

Scheduling

| START ATTACH
| START BREXIT

Security

QUERY SECURITY

Spool Interface (JES)

SPOOLCLOSE
SPOOLOPEN INPUT
SPOOLOPEN OUTPUT
SPOOLREAD
SPOOLWRITE

Storage control

FREEMAIN
GETMAIN

Syncpoint

SYNCPOINT
SYNCPOINT ROLLBACK

Task control

CHANGE TASK
DEQ
ENQ
SUSPEND
WAIT EXTERNAL
WAITCICS

TCP/IP services

| EXTRACT CERTIFICATE
| EXTRACT TCPIP

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 (2770)
CONVERSE (2780)
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
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 (2770)
 RECEIVE (2780)
 RECEIVE (2980)
 RECEIVE (3270 display)
 RECEIVE (3270 logical)
 RECEIVE (3600 BTAM)
 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 (2770)
 SEND (2880)
 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

Web services

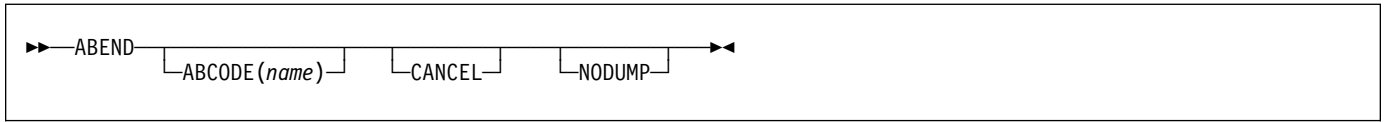
WEB ENDBROWSE FORMFIELD
 | WEB ENDBROWSE HTTPHEADER
 | WEB EXTRACT
 # WEB READ FORMFIELD
 | WEB READ HTTPHEADER
 # WEB READNEXT FORMFIELD
 | WEB READNEXT HTTPHEADER
 | WEB RECEIVE
 | WEB RETRIEVE
 | WEB SEND
 # WEB STARTBROWSE FORMFIELD
 | WEB STARTBROWSE HTTPHEADER
 | WEB WRITE HTTPHEADER

Transient data

DELETEQ TD
 READQ TD
 WRITEQ TD

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.

| ABCODE follows the format rules for DUMPCODE. The
 | EXEC CICS DUMP TRANSACTION command gives the
 | format rules that apply to DUMPCODE, if these rules are
 | not followed, ABEND does not produce a dump.

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 PL/I STAE execution-time option has been specified, an abnormal termination exit is established by PL/I. This exit is revoked by the CANCEL option.

NODUMP

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

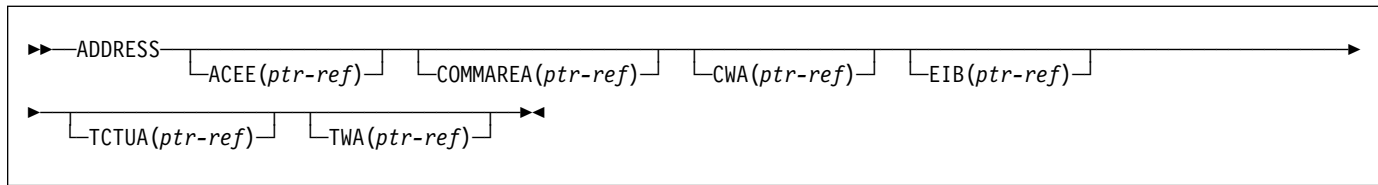
Examples

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.

Note: In OS/VS COBOL only, if an ADDRESS command is included in a COBOL program that is to be compiled using the optimization feature, it must be followed by SERVICE RELOAD statements to reload the BLL cell being used.

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'.

For information on how to map the ACEE data area, see
the mapping macro IHAACEE supplied in
SYS1.MACLIB.

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 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*)

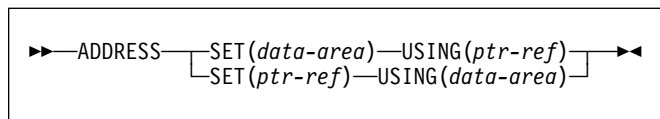
returns a pointer reference, set to the address of the transaction work area (TWA). This area is used for passing information between application programs, but only if they are in the same task. If a TWA does not exist, the pointer reference is set to the null value, X'FF000000'.

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.

ADDRESS SET

Set the address of a structure or pointer.



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)
```

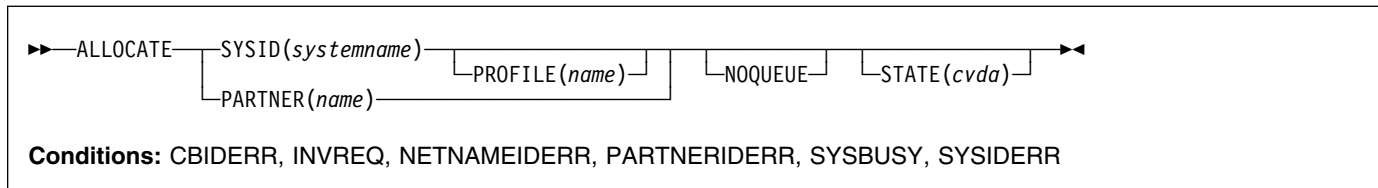
To set a pointer variable to the address of a structure:

```
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.



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

The action taken by CICS if a session is not immediately available depends on whether you specify NOQUEUE (or the equivalent NOSUSPEND option) and also on whether your application has executed a HANDLE command for the SYSBUSY condition. The possible combinations are shown below:

HANDLE for SYSBUSY condition issued

The command is not queued and control is returned immediately to the label specified in the HANDLE command, whether or not you have specified NOQUEUE.

No HANDLE issued for SYSBUSY condition

If you have specified NOQUEUE (or NOSUSPEND), the request is not queued and control is returned immediately to your application program. The SYSBUSY code (X'D3') is set in the EIBRCODE field of the EXEC interface block. You should test this field immediately after issuing the ALLOCATE command.

If you have omitted the NOQUEUE option, CICS queues the request (and your application waits) until a session is available.

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)──┬──PROFILE(name)──┬──NOQUEUE──▶  
              └──SYSID(systemname)──┘
```

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

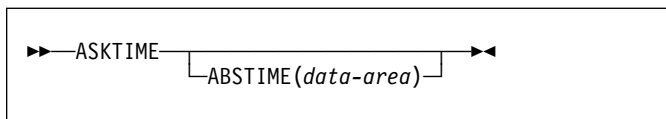
occurs if CICS is unable to provide the application program with a suitable session, for one of the following reasons:

- 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 MVS STCK macro and modifies this by a local time difference. Thus, if your MVS 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 377 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

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

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

Options

ABCODE(*data-area*)

returns a 4-character current abend code. (Abend codes are documented in the *CICS Messages and Codes* 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.

| ACTIVITY(*data-area*)

| returns, if this program is executing on behalf of a CICS
| business transaction services (BTS) activity, the
| 16-character name of the activity.

| BTS is described in the *CICS Business Transaction
| Services* manual.

| ACTIVITYID(*data-area*)

| returns, if this program is executing on behalf of a BTS
| activity, the 52-character, CICS-assigned, identifier of the
| activity-instance.

| BTS is described in the *CICS Business Transaction
| Services* manual.

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, ASRD, or AICA occurred.

The field contains binary zeros if no ASRA, ASRB, ASRD, 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.

ASSIGN

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, ASRD, 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.

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.

ASRASPC(*cvda*)

returns the type of space in control at the time of the last ASRA, ASRB, AICA, or AEYD, abend, if any. The CVDA values on the ASRASPC option are:

SUBSPACE

is returned if the task was executing in either its own subspace or the common subspace at the time of the last ASRA, ASRB, AICA, or AEYD abend.

BASESPACE

is returned if the task was executing in the base space at the time of the last ASRA, ASRB, AICA, or AEYD abend. Note that all tasks execute in base space if transaction isolation is not active.

NOTAPPLIC

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

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 OC4 abend.
- An ASRB or AICA abend has occurred since the last ASRA or AEYD abend.

| BRIDGE(*data-area*)

| returns the 4-character TRANSID of the bridge monitor
| transaction that issued a START BREXIT TRANSID
| command to start the user transaction that issued this
| command. Blanks are returned if:

- The user transaction was not started by a bridge monitor transaction.
- This command was issued by a program started by a distributed program link (DPL) request.

| **Note:** If the START BREXIT command was issued from
| a bridge exit, the TRANSID returned is the that
| of the bridge monitor that issued a START
| BREXIT naming the bridge exit.

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 28) 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 387. 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

ASSIGN

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, a Bridge Exit 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.

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 409 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 associated with the USERID for the current
| task (which could be the default USERID). Refer to the
| SIGNON command for an explanation of how this value
| is derived. (NATLANGINUSE does not show the system
| default language as specified on the NATLANG system
| initialization parameter.)

| See Appendix I, “National language codes” on page 409
| for possible values of the code.

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.

If this command was issued by a user transaction which was started by a Bridge Transaction, the value returned is the termid of the Bridge Facility.

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.

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.

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 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 CONNECTION 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)

| **PROCESS**(*data-area*)

| returns, if this program is executing on behalf of a CICS
| business transaction services (BTS) activity, the
| 36-character name of the BTS process that contains the
| activity.

| BTS is described in the *CICS Business Transaction
| Services* manual.

| **PROCESSTYPE**(*data-area*)

| returns, if this program is executing on behalf of a BTS
| activity, the 8-character process-type of the BTS process
| that contains the activity.

| BTS is described in the *CICS Business Transaction
| Services* manual.

PROGRAM(*data-area*)

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

ASSIGN

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) 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 filed 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 terminal resource definition. This is the DEVICE attribute (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 387.

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.

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:

- 1 The task does not have a signed-on user.
- 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.
- 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.
- 6 A CICS BTS request is issued from outside the CICS BTS environment. (Therefore, it is not executing on behalf of a BTS activity.)
- 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).

▶—BIF DEEDIT—FIELD(*data-area*)—LENGTH(*data-value*)—▶

Condition: LENGERR

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)
```

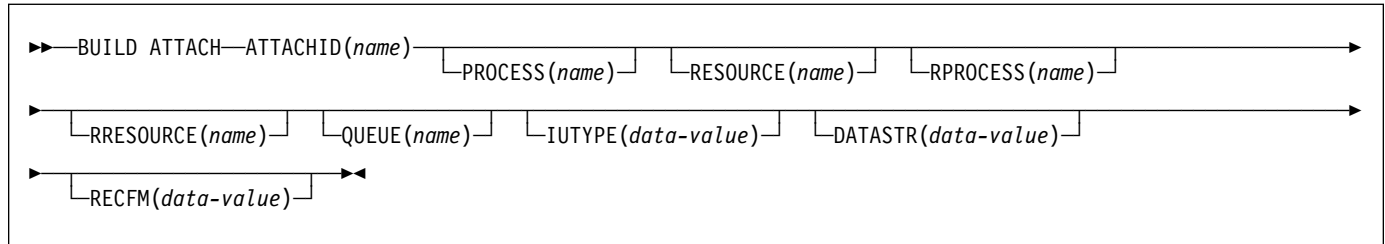
This removes all characters other than digits from CONTG, a 9-byte field, and returns the edited result in that field to the application program. Two examples of the contents of CONTG before and after execution of the command are:

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.

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)

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.

Information about CICS to IMS communication is given in the *CICS Intercommunication Guide*. For details of communication between a CICS system and any other 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:

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*)

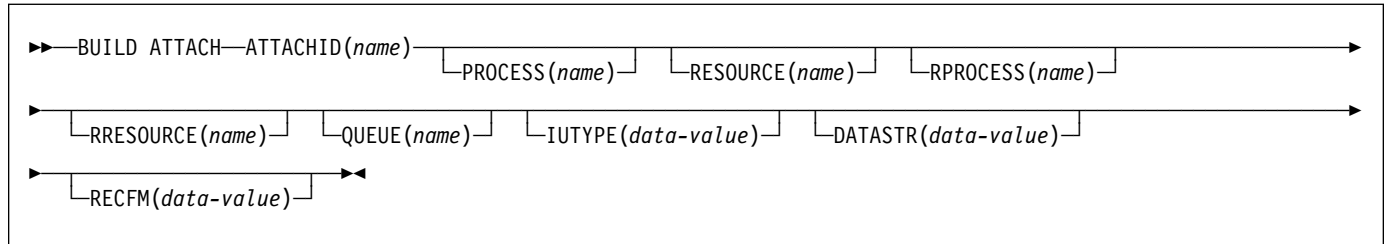
corresponds to the return-resource name, ATTRPRN, in an LUTYPE6.1 attach FMH.

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.

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

BUILD ATTACH (MRO)

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*)

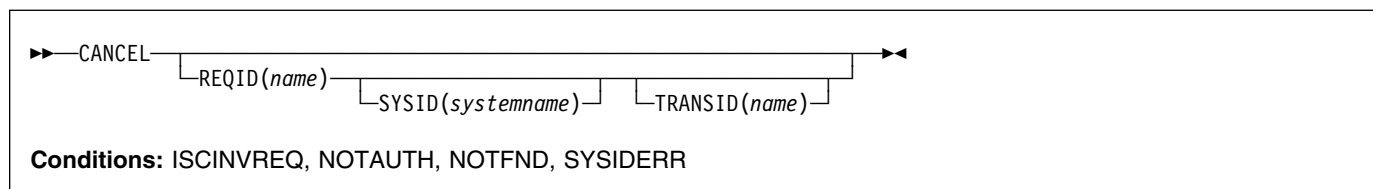
corresponds to the return-resource name, ATTRPRN, in an LUTYPE6.1 attach FMH.

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.



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.

Default action: terminate the task abnormally.

CANCEL**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 a CONNECTION). 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.

```
▶▶—CHANGE PASSWORD(data-value)—NEWPASSWORD(data-value)—USERID(data-value)—└─ESMREASON(data-area)—┘▶
```

```
▶└─ESMRESP(data-area)—▶▶
```

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.

If the ESM is RACF, this field is the RACF reason code.

ESMRESP(*data-area*)

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

If the ESM is RACF, this field is the RACF return code.

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. If the external security manager is RACF, the revoke count maintained by RACF is incremented.
- 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:

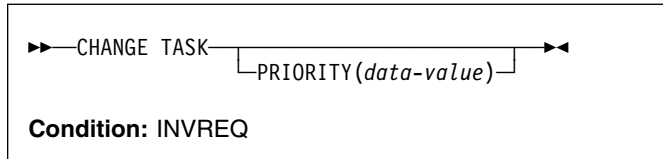
- 8 The USERID is not known to the external security manager.

Default action: terminate the task abnormally.

CHANGE TASK

CHANGE TASK

Change priority of a task.



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

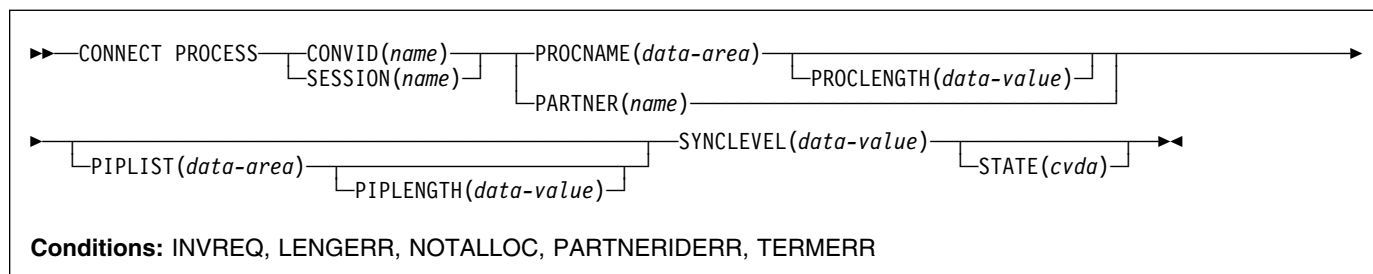
INVREQ

RESP2 values:

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

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
 CONFFREE
 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

CONNECT PROCESS

2 Syncpoint

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.

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 in use by CPI-Communications or that is an APPC basic conversation. In the latter case, GDS CONNECT PROCESS should have been used.

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 PIPELENGTH option is less than 0.
- The value specified in the PIPELENGTH 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 PIPELENGTH.

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.

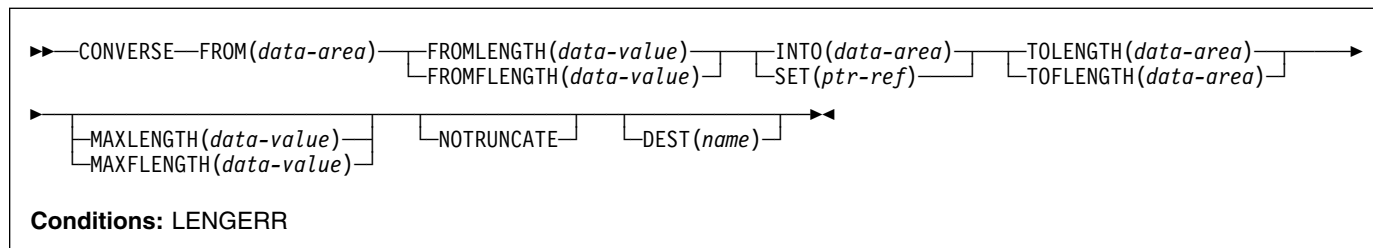
TERMERR

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

CONVERSE (VTAM default)

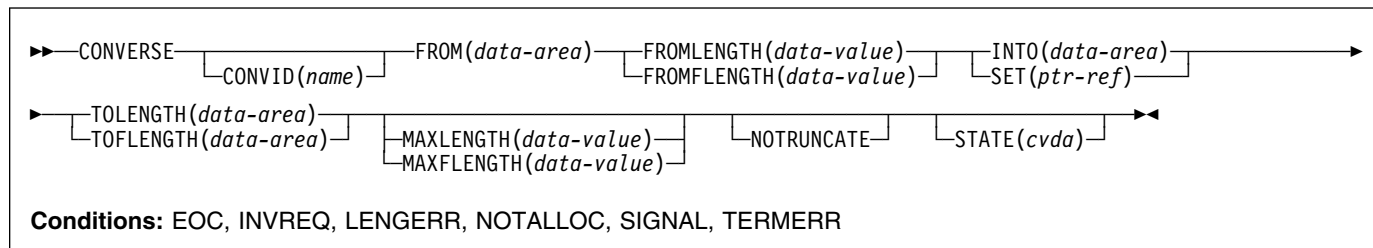
Communicate on standard CICS terminal support.



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.



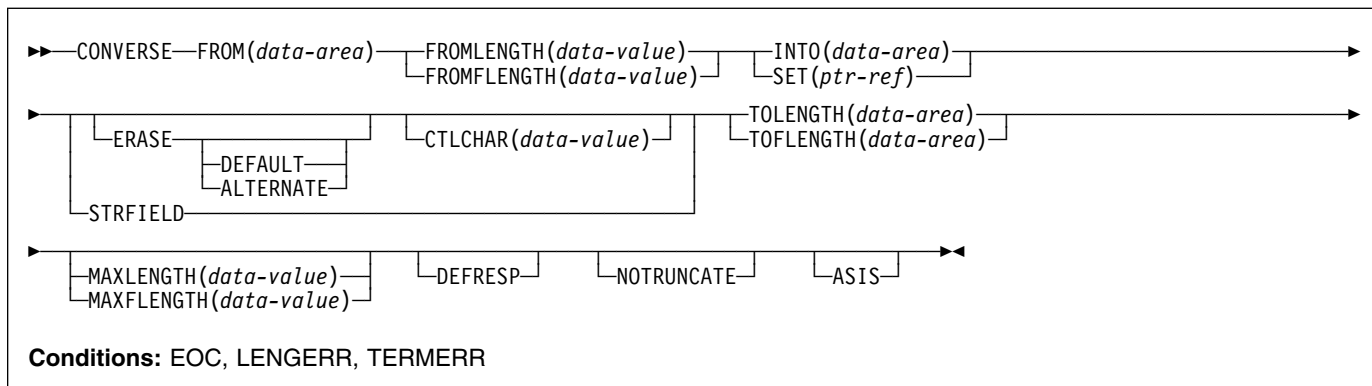
Description

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

CONVERSE (VTAM)

CONVERSE (LUTYPE2/LUTYPE3)

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

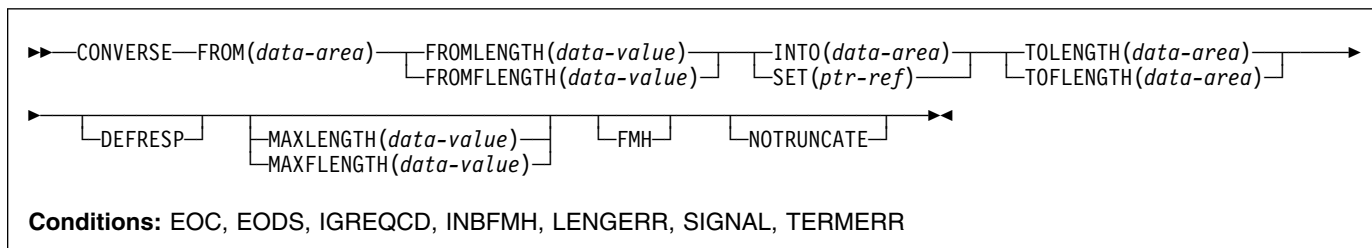


Description

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

CONVERSE (LUTYPE4)

Communicate on an LUTYPE4 logical unit.

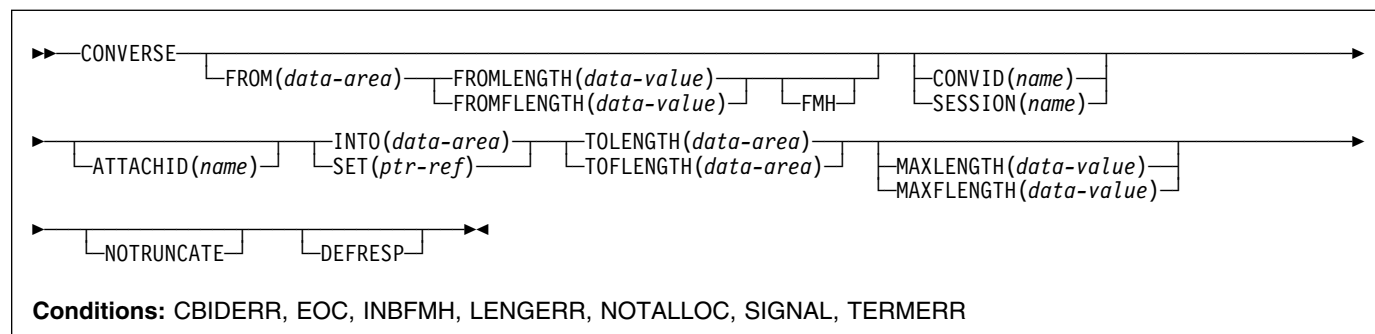


Description

CONVERSE communicates on an LUTYPE4 logical unit.

CONVERSE (LUTYPE6.1)

Communicate on an LUTYPE6.1 logical unit.

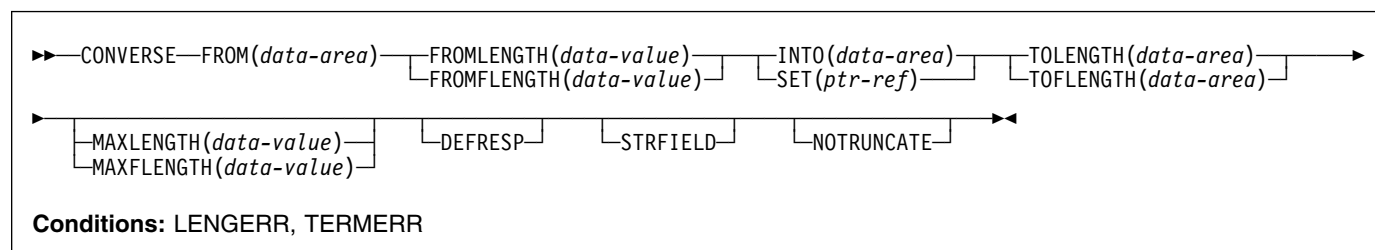


Description

CONVERSE communicates on an LUTYPE6.1 logical unit.

CONVERSE (SCS)

Communicate on a 3270 SCS printer logical unit.



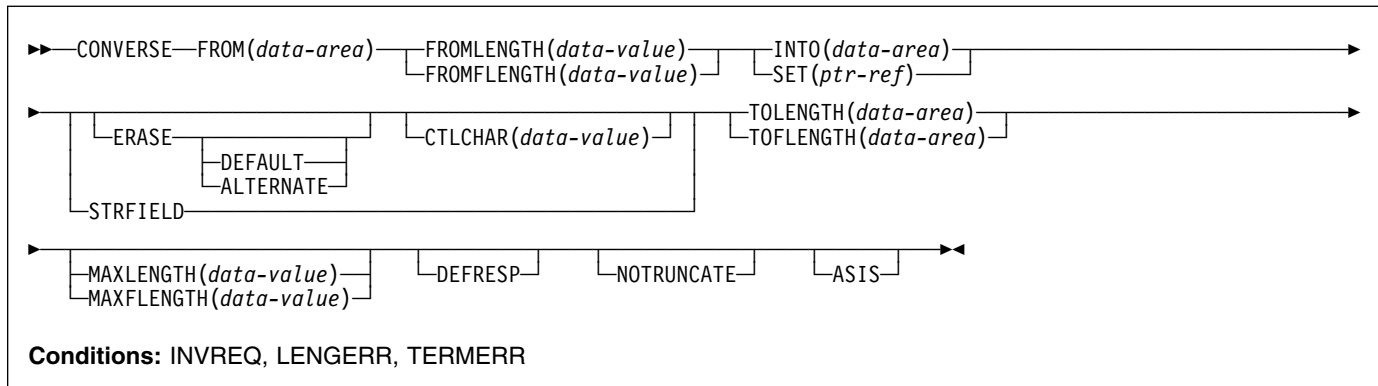
Description

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 (VTAM)

CONVERSE (3270 logical)

Communicate on a 3270 logical unit.

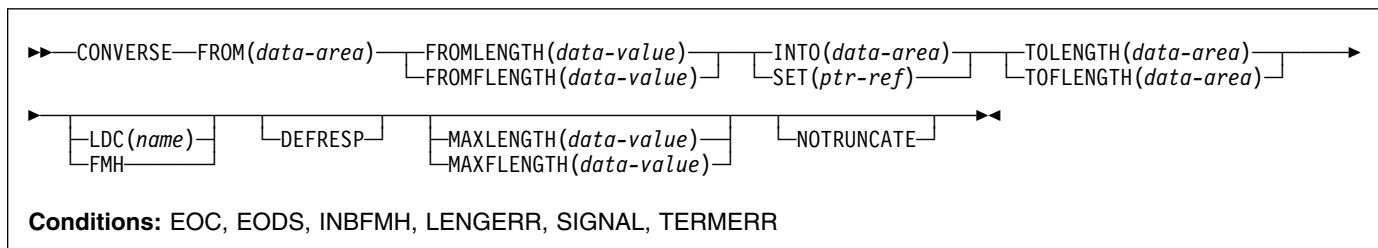


Description

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 4770 and 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.

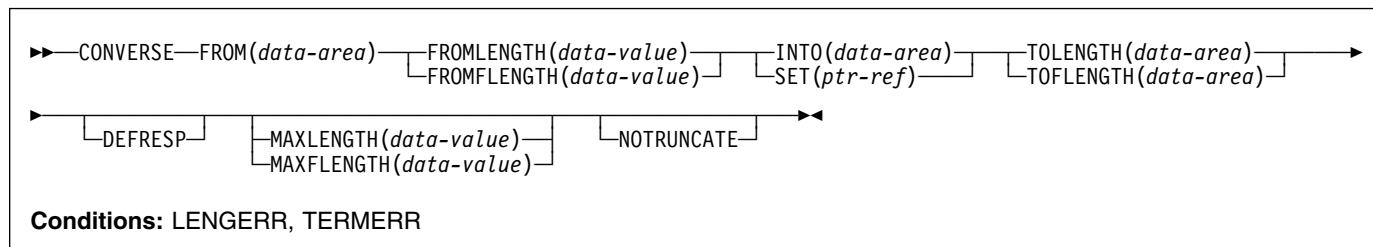
The installation can specify up to 256 2-character mnemonics for each TCTTE, and two or more TCTTEs can share a list of these mnemonics. A numeric value (0 through 255) corresponds to each LDC mnemonic for each TCTTE.

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.

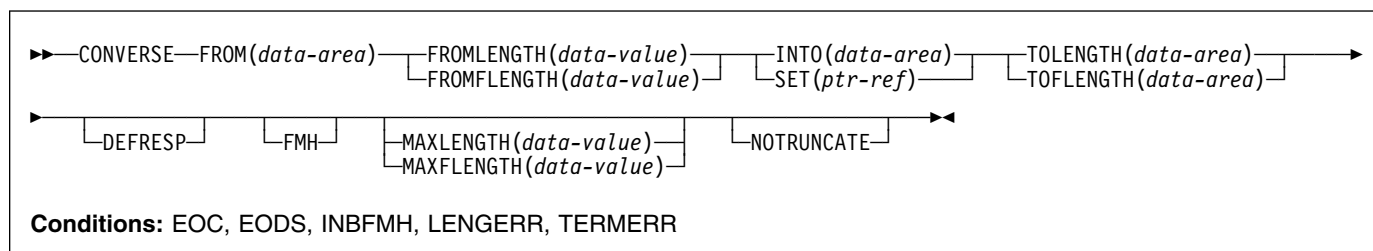
The data stream and communication format used between a CICS application program and a 3614 is determined by the 3614. The application program is therefore device_dependent when handling 3614 communication.

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

On output, the FMH can be built by the application program or by CICS. If your program supplies the FMH, you place it at the # front of your output data and specify the FMH option on your CONVERSE command. If you omit the FMH option, CICS will # provide an FMH but you must reserve the first three bytes of the message for CICS to fill in.

CONVERSE (3650 interpreter)

Communicate on a 3650 interpreter logical unit.



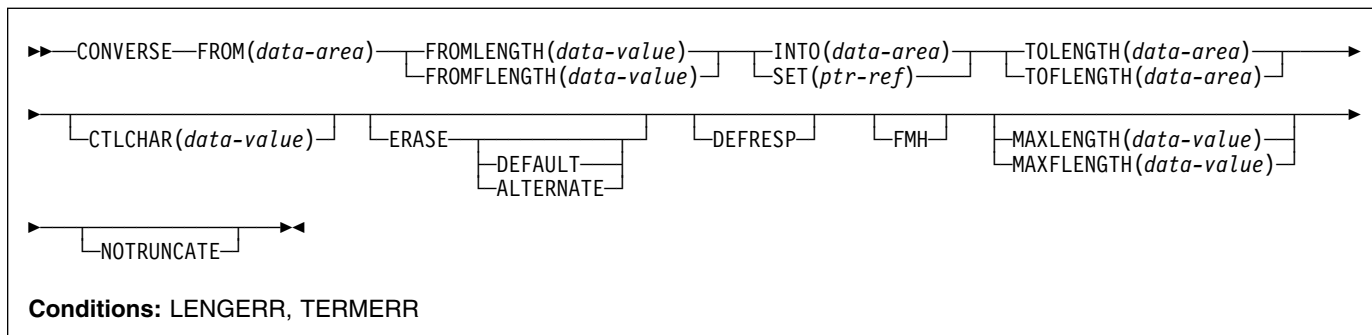
Description

CONVERSE communicates on a 3650 interpreter logical unit.

CONVERSE (VTAM)

CONVERSE (3650-3270)

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

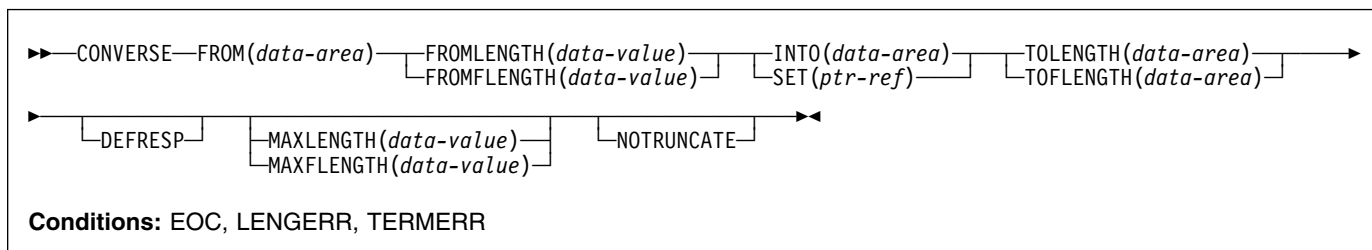


Description

CONVERSE communicates on a 3650 host conversational logical unit.

CONVERSE (3650-3653)

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

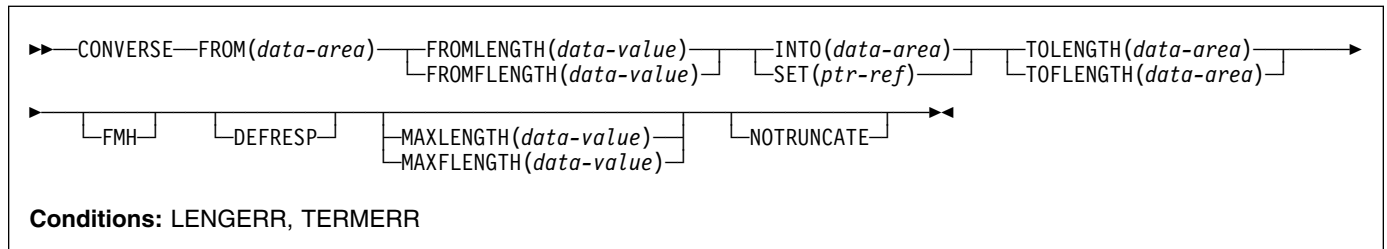


Description

CONVERSE communicates on a 3650 host conversational logical unit.

CONVERSE (3650-3680)

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

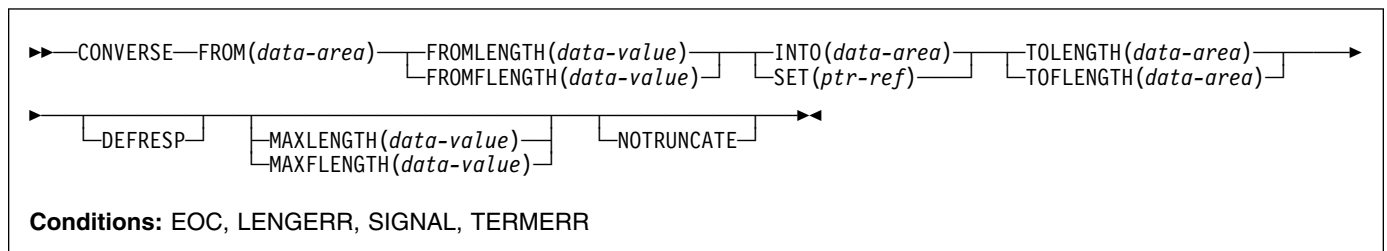


Description

CONVERSE communicates on a 3650 host command processor logical unit.

CONVERSE (3767)

Communicate on a 3767 interactive logical unit.



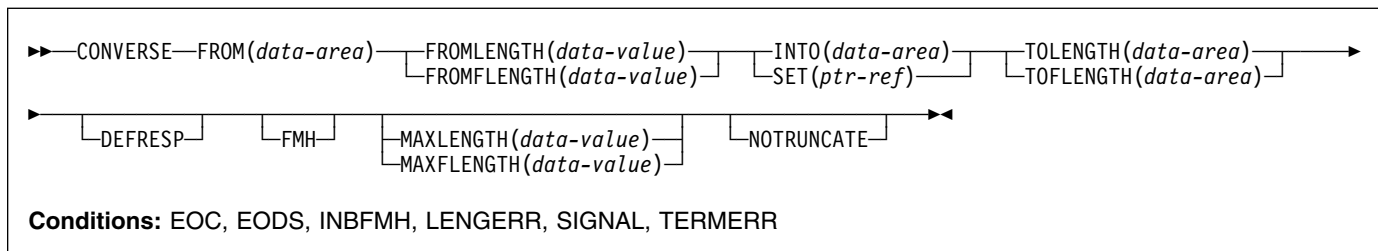
Description

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

CONVERSE (VTAM)

CONVERSE (3770)

Communicate on a 3770 batch logical unit.

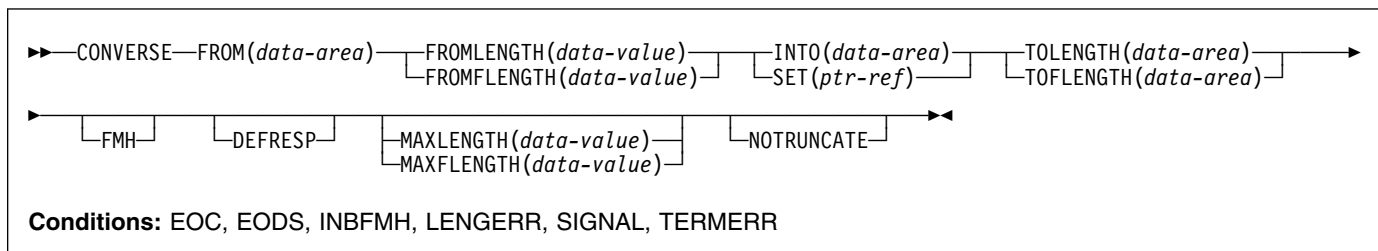


Description

CONVERSE communicates on a 3770 batch logical unit.

CONVERSE (3790 full-function or inquiry)

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

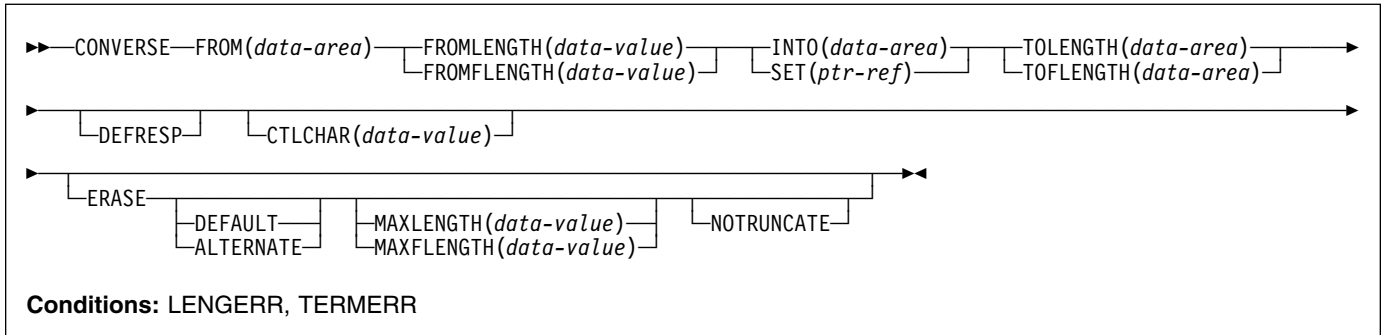


Description

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.

CONVERSE (VTAM)

Options

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.

DEST(*name*)

specifies the 4-byte symbolic name of the TCAM destination to which the message is to be sent. This option is meaningful only for terminals defined using DFHTCT TYPE=SDSCI with DEVICE=TCAM.

If you use the DEST option, you must be aware of any restrictions placed on device-dependent data streams by the message control facility in use. See the section on CICS-TCAM interface in the *CICS Customization Guide* for programming information.

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.

The use of FMH is optional and is not supported for all
terminal types. If not supplied, CICS takes no action,
except for 3600/4700 terminals, where an FMH is
mandatory. In this case, if FMH is not specified, CICS
supplies one and places it in the first 3 bytes of the
message, which you must reserve for this purpose.

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.

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.

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
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 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.

CONVERSE (VTAM)

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.

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 NOTRUNCATE 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

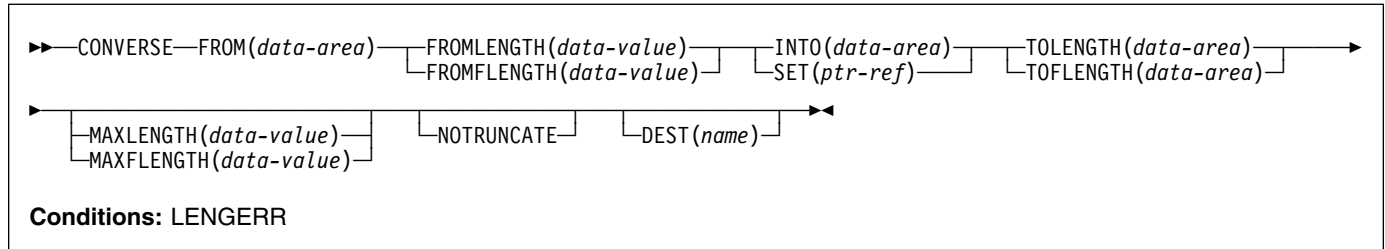
occurs for a terminal or session-related error. Any action on that conversation other than a FREE causes an ATCV abend.

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 default)

Communicate on standard CICS terminal support.

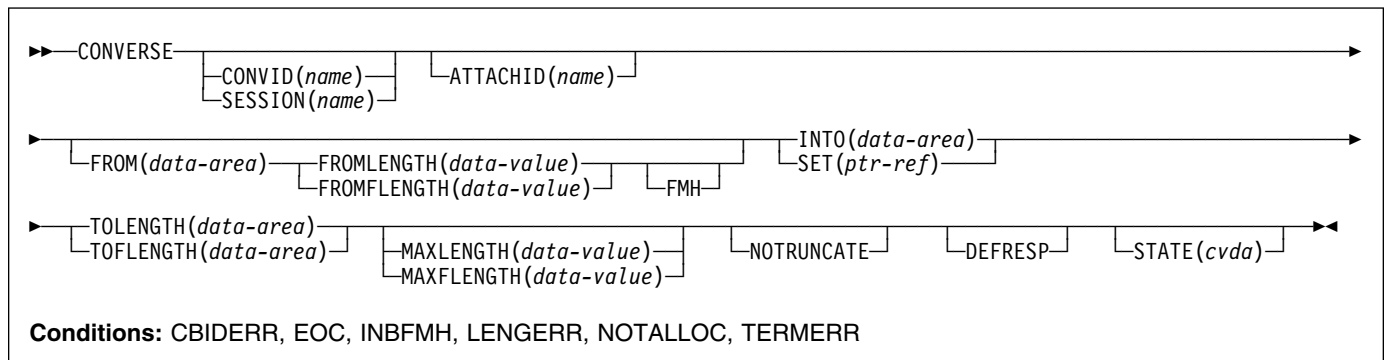


Description

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.

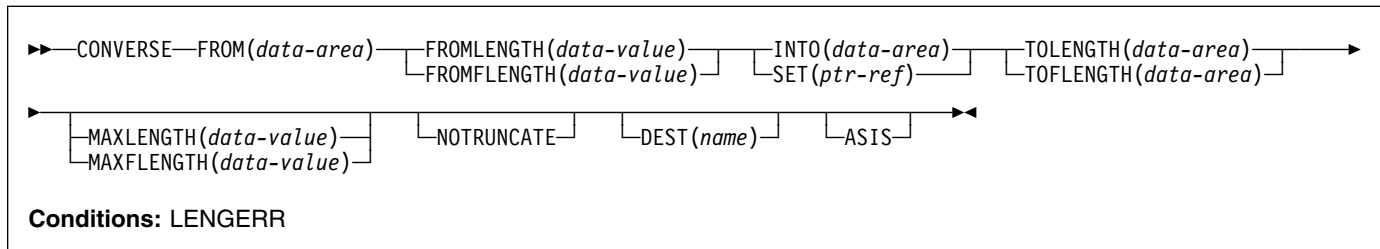


CONVERSE communicates on an MRO session. For more information about MRO and IRC, see the *CICS Intercommunication Guide*.

CONVERSE (non-VTAM)

CONVERSE (System/3)

Communicate on a System/3 terminal.



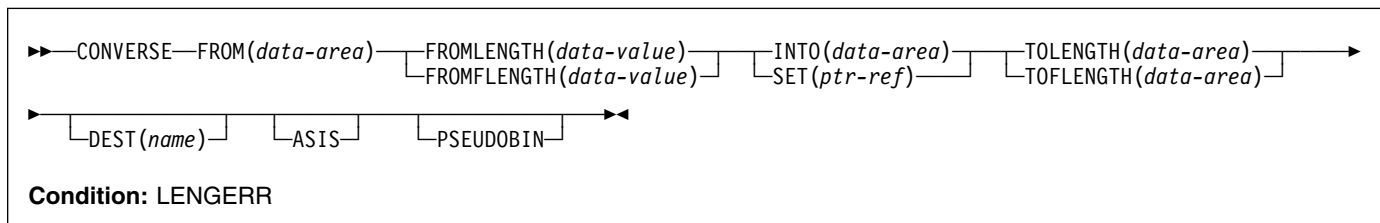
Description

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.

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.

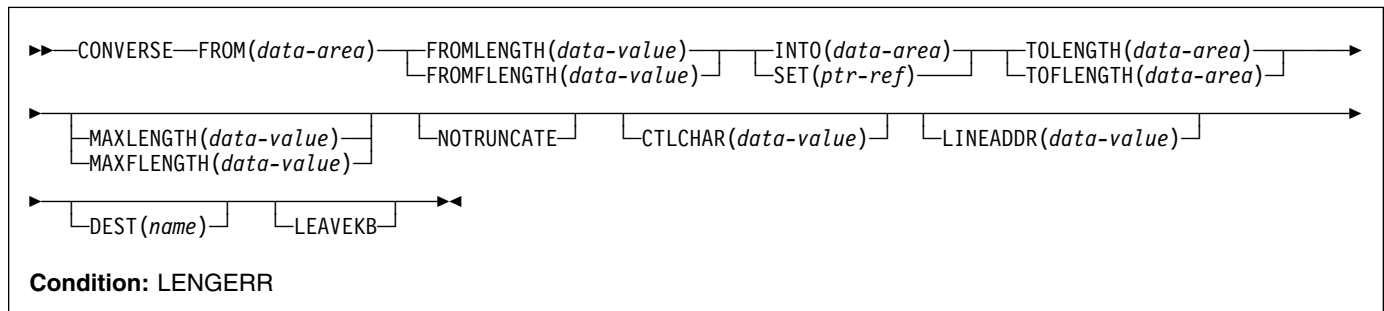
When a disconnect is issued to a dial-up System/7, the "busy" bit is sometimes left on in the interrupt status word of the ACCA. If the line connection is reestablished by dialing from the System/7 end, the 'busy' condition of the ACCA prevents message transmission from the System/7. To overcome this 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)

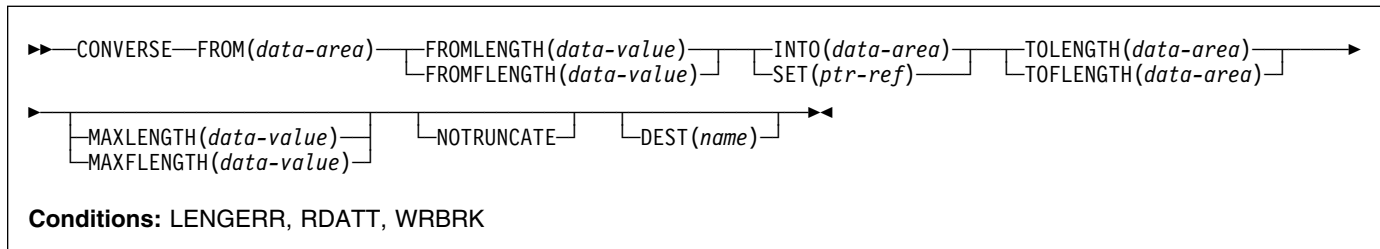
Communicate on a 2260 or 2265 display station.



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.

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 54 .

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)

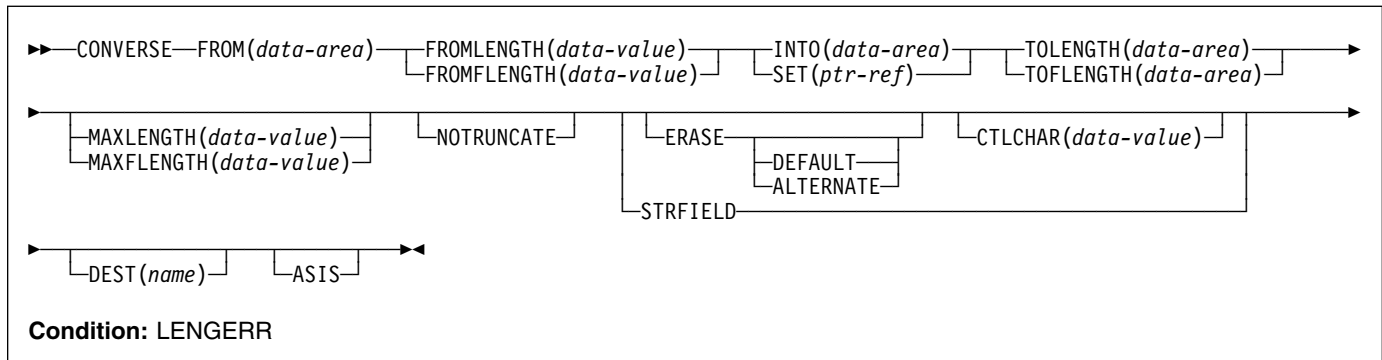
The support and command syntax for 2780 is the same as for System/3, see “CONVERSE (System/3)” on page 54.

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 (non-VTAM)

CONVERSE (3270 display)

Communicate on a 3270 information display system (TCAM).

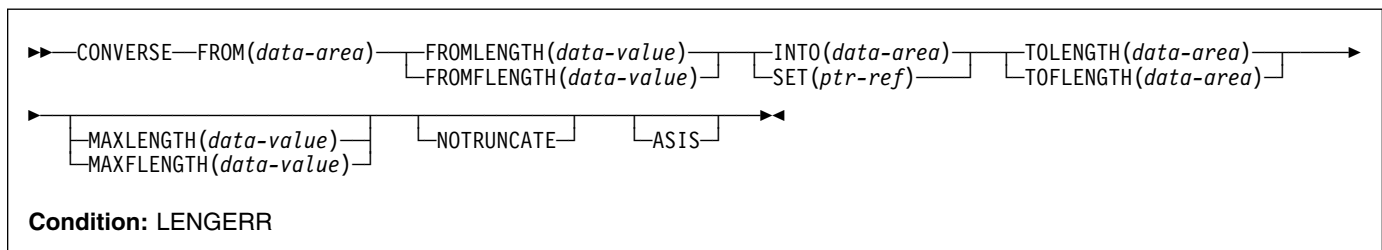


Description

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 394.

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.

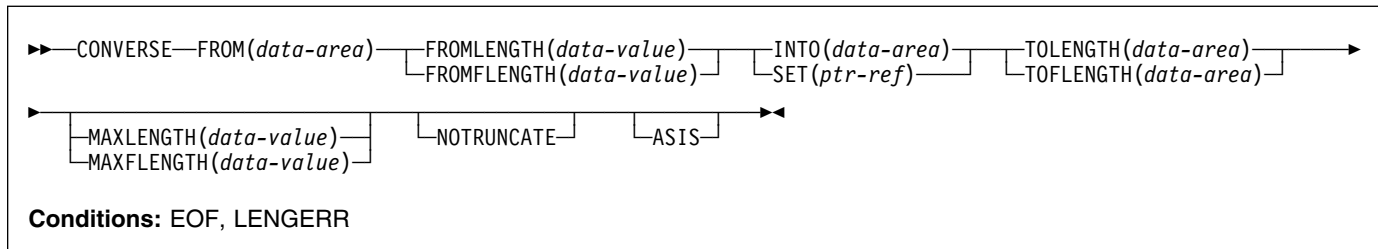
Only one resend message at a time can be queued for a logical unit. If a second resend situation occurs before CICS has written the first, a resend message is sent containing the eight bytes of data that accompanied the second input transaction from the 3600 logical unit.

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 (non-VTAM)

CONVERSE (3735)

Communicate on a 3735 programmable buffered terminal.

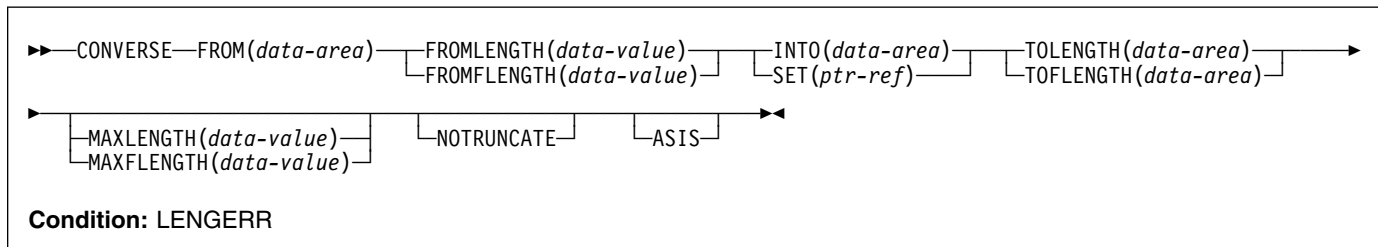


Description

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.

Options

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.

DEST(*name*)

specifies the 4-byte symbolic name of the TCAM destination to which the message is to be sent. This option is meaningful only for terminals defined using DFHTCT TYPE=SDSCI with DEVICE=TCAM.

If you use the DEST option, you must be aware of any restrictions placed on device-dependent data streams by

the message control facility in use. See the section on the CICS-TCAM interface in the *CICS Customization Guide* for programming information.

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,

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

CONVERSE (non-VTAM)

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.

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.

TOLENGTH(*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 (not TCAM)

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.

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 (not TCAM)

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.

DEFINE COUNTER

DEFINE COUNTER

Define a named counter.

```
DEFINE COUNTER(name) [POOL(name)] [VALUE(data-value)] [MINIMUM(data-value)]  
[MAXIMUM(data-value)]
```

Conditions: INVREQ

```
DEFINE DCOUNTER(name) [POOL(name)] [VALUE(data-area)] [MINIMUM(data-area)]  
[MAXIMUM(data-area)]
```

Conditions: INVREQ

Description

These counter commands create a new named counter in a named counter pool in the coupling facility.

DEFINE COUNTER creates counters that are handled as fullword *signed* binary values, and DEFINE DCOUNTER creates counters that are handled as doubleword *unsigned* binary values.

Note: Although the CICS API allows you to operate with either fullword (signed) or doubleword (unsigned) binary values, the named counter server stores *all* values as doubleword unsigned values. This can give rise to overflow conditions if, for example, you define a counter with the DCOUNTER command and try to access it using the COUNTER command. You should always access a named counter using commands from the *same command set* that you used to define the counter.

For information about specifying fullword and doubleword variables on these named counter commands, see "Argument values" on page 2.

Options

COUNTER(*name*)

specifies the 16-character name of the named counter to be created. All value fields for this counter are handled as fullword signed binary values. Valid characters for names are A through Z, 0 through 9, \$ @ # and _ (underscore). If *name* is a variable that contains a name that is less than 16 characters, the name must be padded with trailing blanks.

DCOUNTER(*name*)

specifies the 16-character name of the named counter to be created. All value fields for this counter are handled as doubleword unsigned binary values. Valid characters for names are A through Z, 0 through 9, \$ @ # and _ (underscore). If *name* is a variable that contains a name that is less than 16 characters, the name must be padded with trailing blanks.

MAXIMUM(*data-value*)

specifies the maximum number for the named counter, using a fullword signed binary value for COUNTER and a doubleword unsigned value for DCOUNTER. This is the maximum number that can be assigned on a GET command, after which the counter must be reset by a REWIND command.

If you omit the MAXIMUM parameter, the named counter is defined with a default maximum of high-values (X'7FFFFFFF' for the signed fullword case, or a doubleword filled with X'FF').

MINIMUM(*data-value*)

specifies the minimum number for the named counter, using a fullword signed binary value for COUNTER and a doubleword unsigned value for DCOUNTER. This is the value to which a named counter is reset as a result of a REWIND command.

If you omit the MINIMUM parameter, the named counter is defined with a default minimum of low-values (a fullword or doubleword filled with X'00').

POOL(*name*)

specifies an 8-character string to be used as a pool selection parameter to select the pool in which the named counter is to be created. The string can be a logical pool name, or the actual pool name.

| Valid characters for the pool selector string are A through Z, 0 through 9, \$ @ # and _ (underscore). If *name* is a variable that contains a name that is less than 8 characters, the name must be padded with trailing blanks.

| This parameter is optional. If you omit the name of the pool, a pool selector value of 8 blanks is assumed.

| If there is no matching entry in the DFHNCOPT options table, CICS uses the name specified on the NCPLDFT system initialization parameter, which specifies the default named counter pool.

| For information about generating a named counter options table using the DFHNCO macro, see the *CICS System Definition Guide*.

| **VALUE**(*data-value*)

| specifies the initial number at which the new named counter is to start, using a fullword signed binary value for COUNTER and a doubleword unsigned value for DCOUNTER.

| You can specify a number that is equal to, or greater than, the minimum value, up to the maximum value plus 1. If you specify an initial number that is equal to the maximum value plus 1, the counter is created with the counter-at-limit condition set and it cannot be used until it is rewind.

| If you omit both the VALUE and MINIMUM parameters, the named counter is created with an initial value of zero.

| If you omit VALUE but specify a MINIMUM , the translator issues an error—the VALUE parameter is required if you specify the MINIMUM parameter.

| **Conditions**

| **INVREQ**

| RESP2 values:

- | **202** Duplicate counter name. A named counter of this name already exists.
- | **301** The server has reported an error code that is not understood by the named counter interface. Generally, this is not possible unless the interface load module, DFHNCIF, is at a lower maintenance or release level than the server itself.
- | **302** The server cannot create the new named counter because there is not enough space in the named counter pool.
- | **303** An unexpected error, such as structure failure or loss of connectivity, has occurred on a macro used to access the coupling facility. Further information can be found in message DFHNC0441 in the CICS job log.
- | **304** The pool selection parameter specified in the program cannot be resolved to a valid server name using the current options table.

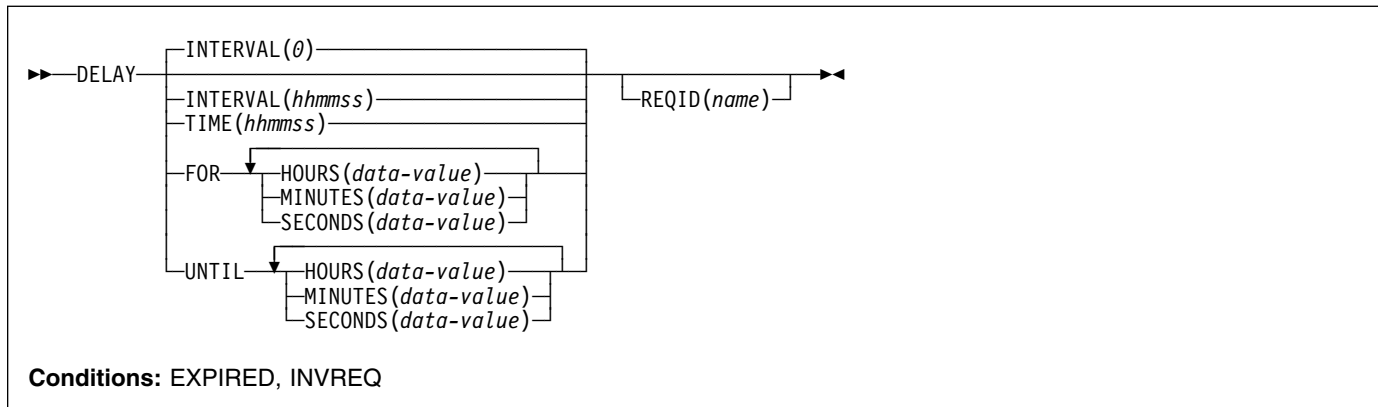
- | **305** The interface is unable to establish a connection to the server for the selected named counter pool. Further information can be found in an AXM system services message (AXMSCnnnn) in the CICS job log.
- | **306** An abend occurred during server processing of a request. Further information can be found in a message in the CICS job log and the server job log.
- | **308** The DFHNCOPT options table module, required for resolving a pool name, could not be loaded.
- | **309** During processing of the options table, the named counter interface encountered an unknown entry format. Either the options table is not correctly generated, or the DFHNCIF interface load module is not at the same release level as the options table.
- | **310** An options table entry matching the given pool name specified a user exit program, but the user exit program is not link-edited with the options table and cannot be loaded.
- | **403** The POOL parameter contains invalid characters or embedded spaces.
- | **404** The COUNTER parameter contains invalid characters or embedded spaces.
- | **406** The VALUE parameter is invalid. Initial values cannot be less than the minimum value, and cannot be greater than the maximum value plus 1.
- | **407** The MINIMUM or MAXIMUM parameter is invalid. Either the MAXIMUM parameter specifies a value that is less than the minimum value, or (for COUNTER only) one of the parameters specifies a negative value.

| Default action: terminate the task abnormally.

DELAY

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 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*.

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')
```

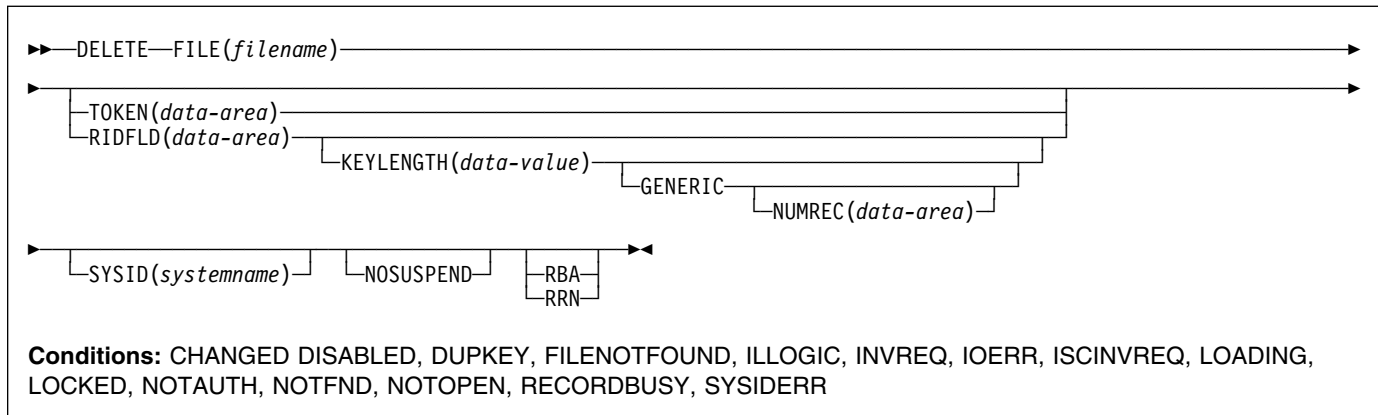
There are two ways to enter the time under FOR or UNTIL.

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

DELETE

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 BDAM file. **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.

When this command is used to delete records from a coupling facility data table, the update is made only to the data table in the coupling facility.

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.

NOSUSPEND (RLS only)

specifies that the request is not to wait if VSAM is holding an active lock against the record, including records locked as the result of a DEADLOCK.

NUMREC(*data-area*) (VSAM KSDS only)

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

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
- Coupling facility data tables
- Any files opened in RLS access mode
- 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 or coupling facility 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 or BROWSE for UPDATE request. The value to use is the value returned in the TOKEN held by the earlier READ UPDATE or BROWSE for UPDATE request.

TOKEN can be function shipped. However, if a request specifying TOKEN is function shipped to a member of the CICS family of products that does not recognize this option, the request fails.

Conditions

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

CHANGED

RESP2 values:

109 A DELETE command (without RIDFLD) is issued for a file that is defined as a coupling facility data table using the contention update model and the record has been changed since the application program read it for update. To perform the DELETE successfully, repeat the read for update to get the latest version of the record, and try the DELETE again.

Default action: terminate the task abnormally.

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 377 for details.)

Default action: terminate the task abnormally.

INVREQ

RESP2 values:

DELETE

- 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.
- 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 BDAM 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 or coupling facility 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.
- 51 A DELETE command specifying the RBA keyword is issued against a KSDS file that is being accessed in RLS mode. RLS does not support relative byte address (RBA) access to KSDS files.
- 55 NOSUSPEND is specified for a non-RLS file.
- 56 An attempt to update a recoverable coupling facility data table has failed because the current unit of work has already updated 1024 recoverable coupling facility data tables. You cannot update more than 1024 recoverable coupling facility data tables within a unit of work

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.
- For a coupling facility data table, an IOERR indicates a bad response returned from a coupling facility access.

(Further information is available in the EXEC interface block; refer to Appendix A, "EXEC interface block" on page 377 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 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.

LOADING is also returned for a coupling facility data table if the delete request is for a key that is not yet loaded. A coupling facility data table can be modified during loading, but only if the requested key is within the range of those records already loaded.

The LOADING response can also be returned for a coupling facility data table that has failed during loading. For more information about what happens if the load for a coupling facility data table fails, see the description of the XDTLC global user exit in the *CICS Customization Guide*.

If your application programs encounter the LOADING condition persistently or too frequently, check that this is not caused by conflicting file definitions that reference the same data set.

Default action: terminate the task abnormally.

LOCKED

RESP2 values:

- 106 An attempt is made to delete a record specifying the RIDFLD, but a *retained* lock exists against this key (see "Retained and active locks" on page 71). If the request specifies the GENERIC keyword, all possible records are deleted, but the locked records remain. The number of records deleted is returned by NUMREC.

The LOCKED condition can also occur for a DELETE request to a recoverable CFDT that uses the locking model, if the record being read is locked by a retained lock. See the *CICS Recovery and Restart Guide* for information about investigating retained locks on records in a coupling facility data table.

Default action: abend the task with code AEX8.

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 and coupling facility 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 (if the table was created from one); 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 data table.

| For coupling facility data tables, this condition can
 | also occur when a DELETE command (without a
 | RIDFLD) is issued to a coupling facility data table
 | using the contention model, and the record has
 | been deleted since it was read for update.

Default action: terminate the task abnormally.

NOTOPEN

RESP2 values:

- 60** NOTOPEN (RESP2 60) is returned for one of the following reasons:
 - 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. You can also make CLOSED, UNENABLED the initial state, by specifying STATUS(UNENABLED) and OPENTIME(FIRSTREF) on the FILE resource definition.
 - The requested file is OPEN and in use by other transactions, but a CLOSE request against the file has been received.
 - A DELETE command is issued against a data set that is quiesced, or is being quiesced, as a result of a SET DSNAME QUIESCED or IMMQUIESCED command.

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.

RECORDBUSY

RESP2 values:

- 107** The NOSUSPEND keyword is specified for the deletion of a record that is locked by a VSAM active lock (see "Retained and active locks").

If the request specifies the GENERIC keyword, all possible records are deleted except for the locked records which remain. The number of records deleted is returned by NUMREC.

Default action: abend the task with code AEX9.

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.
- 131** For a coupling facility data table, the connection to the coupling facility data table server has failed. This could be because the server itself has failed, or the server is available, but CICS has failed to connect to it.
- 132** The DELETE is issued against a coupling facility data table that no longer exists, probably because of a coupling facility failure, in which case the coupling facility data table server also fails. See the *CICS System Definition Guide* for information about restarting a coupling facility data table server and reloading a table.

Default action: terminate the task abnormally.

Description

Retained and active locks: RECORDBUSY refers to active locks and LOCKED refers to retained locks:

- DELETE requests for records that have *retained* locks are always rejected with a LOCKED response.
- DELETE requests for records that have *active* locks wait for the lock to be released, except when the NOSUSPEND keyword is specified, in which case CICS returns the RECORDBUSY response.

DELETE

Examples

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

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

DELETE COUNTER

Delete the named counter

```
▶▶DELETE COUNTER(name)—┐POOL(name)┘▶▶
```

Conditions: INVREQ

```
▶▶DELETE DCOUNTER(name)—┐POOL(name)┘▶▶
```

Conditions: INVREQ

Description

These commands delete the named counter from the specified pool. COUNTER operates on a fullword counter, and DCOUNTER operates on a doubleword counter.

Options

COUNTER(*name*)

specifies the name of the fullword counter to be deleted. The name can be up to 16 alphanumeric characters. If *name* is a variable that contains a name that is less than 16 characters, the name must be padded with trailing blanks.

DCOUNTER(*name*)

specifies the name of the doubleword counter to be deleted. The name can be up to 16 alphanumeric characters. If *name* is a variable that contains a name that is less than 16 characters, the name must be padded with trailing blanks.

POOL(*poolname*)

specifies an 8-character string to be used as a pool selection parameter to select the pool in which the named counter resides. The string can be a logical pool name, or the actual pool name.

Valid characters for the pool selector string are A through Z, 0 through 9, \$ @ # and _ (underscore). If *name* is a variable that contains a name that is less than 8 characters, the name must be padded with trailing blanks.

This parameter is optional. If you omit the name of the pool, a pool selector value of 8 blanks is assumed.

If there is no matching entry in the DFHNCOPT options table, CICS uses the name specified on the NCPLDFT system initialization parameter, which specifies the default named counter pool.

For information about generating a named counter options table using the DFHNCO macro, see the *CICS System Definition Guide*.

Conditions

INVREQ

RESP2 values:

- 201** Named counter not found.
- 301** The server has reported an error code that is not understood by the named counter interface. Generally, this is not possible unless the interface load module, DFHNCF, is at a lower maintenance or release level than the server itself.
- 303** An unexpected error, such as structure failure or loss of connectivity, has occurred on a macro used to access the coupling facility. Further information can be found in message DFHNC0441 in the CICS job log.
- 304** The pool selection parameter specified in the program cannot be resolved to a valid server name using the current options table.
- 305** The interface is unable to establish a connection to the server for the selected named counter pool. Further information can be found in an AXM services message (AXMSCnnnn) in the CICS job log.
- 306** An abend occurred during server processing of a request. Further information can be found in a message in the CICS job log and the server job log.
- 308** The DFHNCOPT options table module, required for resolving a pool name, could not be loaded.
- 309** During processing of the options table, the named counter interface encountered an unknown entry format. Either the options table is not correctly

DELETE COUNTER

- | generated, or the DFHNCIF interface load module
- | is not at the same release level as the options
- | table.
- | **310** An options table entry matching the given pool
- | name specified a user exit program, but the user
- | exit program is not link-edited with the options
- | table and cannot be loaded.
- | **403** The pool selection parameter contains characters
- | that are not allowed, or embedded spaces.
- | Default action: terminate the task abnormally.

DELETEQ TD

Delete all transient data.

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

Conditions: DISABLED, INVREQ, ISCINVREQ, LOCKED, 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 be defined to CICS.

If SYSID is specified, the queue is assumed to be on a remote system, irrespective of how it is defined. Otherwise the resource definition 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.

ISCINVREQ

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

Default action: terminate the task abnormally.

LOCKED

occurs when the request cannot be performed because use of the queue has been restricted owing to a unit of work failing in-doubt. This can happen on any request for a logically-recoverable queue defined with

WAIT(YES) and WAITACTION(REJECT) in the TDQUEUE resource definition.

Specify WAIT(YES) and WAITACTION(Queue) in the TDQUEUE resource definition if you want the transaction to wait.

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 a 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)─┐
                └─QNAME(name)─┘ ┌─SYSID(systemname)─┐▶▶
```

Conditions: INVREQ, ISCVREQ, LOCKED, 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 characters) of the queue to be deleted. The name may not consist solely of binary zeros and must be unique within the CICS system.

QNAME(*name*)

an alternative to QUEUE, QNAME specifies the symbolic name (1–16 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.

LOCKED

occurs when the request cannot be performed because use of the queue has been restricted owing to a unit of work failing in-doubt.

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:

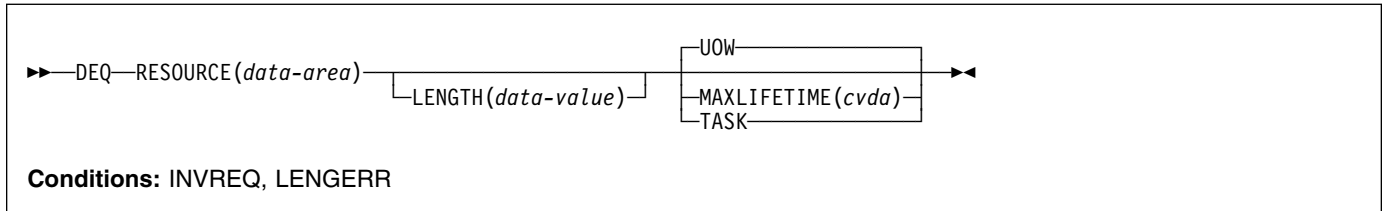
- When the SYSID option specifies a name that is neither the local system nor a remote system (made known to CICS by defining a CONNECTION)
- When the link to the remote system is closed
- When the CICS region in which the temporary storage command is executed fails to connect to the TS server managing the TS pool that supports the referenced temporary storage queue. (For example, this can happen if the CICS region is not authorized to access the temporary storage server).

SYSIDERR can also occur if the temporary storage server has not been started, or because the server has failed (or been stopped) while CICS continues executing.

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, unless the name specified in RESOURCE matches the name specified in an installed ENQMODEL resource definition, that has sysplex-wide scope. 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.

When an EXEC CICS DEQ (or ENQ) command is issued for a resource whose name matches that of an installed ENQMODEL resource definition, CICS checks the value of the ENQSCOPE attribute to determine whether the scope is local or sysplex-wide. If the ENQSCOPE attribute is left blank (the default value), CICS treats the DEQ as local to the issuing CICS region. If no ENQMODEL matches the resource name, the scope of the DEQ command is local. See the *CICS Resource Definition Guide* for more information about the ENQMODEL resource definition.

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:

UOW

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

Note: For compatibility with previous releases of CICS, a CVDA value of LUW is also supported.

TASK

The 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:

- The MAXLIFETIME option is set with an incorrect CVDA.

Default action: terminate the task abnormally.

LENGERR

RESP2 values:

DEQ

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

Default action: terminate the task abnormally.

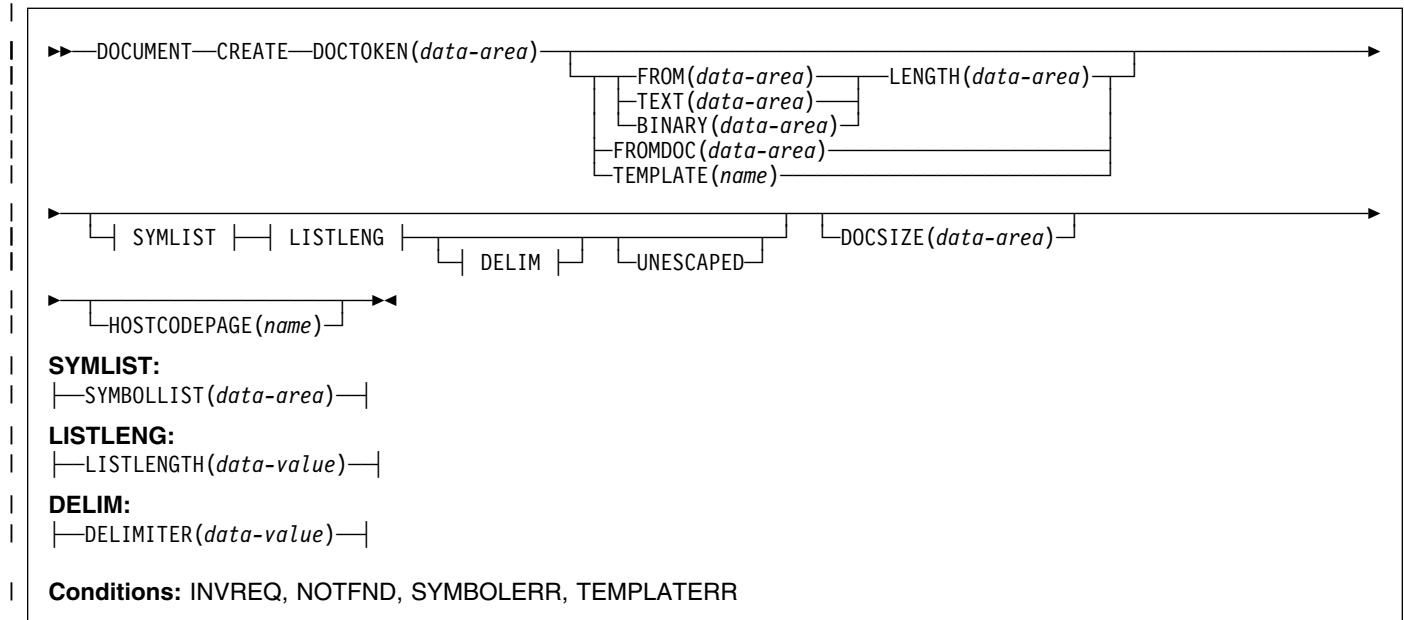
Examples

The following examples show how to dequeue from a resource:

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

DOCUMENT CREATE

Create a document.



Description

DOCUMENT CREATE signals the start of the document creation process. The document being created can be an empty document, or it can be based on an existing document, a template, or data contained in an application buffer.

Options

BINARY(data-area)

specifies a buffer of data which is to be used as the contents of the new document being created. The data is copied unchanged to the document content and no attempt is made to parse the data for symbol substitution. The purpose of the BINARY option is to allow the application to insert blocks of data that must not undergo conversion to a client code page when the data is sent.

DELIMITER(data-value)

specifies an optional 1-byte value used to delimit symbol name-value pairs in the SYMBOLLIST buffer. If this option is not specified, the value defaults to an ampersand. There are several disallowed DELIMITER values, all of which cause an INVREQ condition if used. The disallowed values are:

- # • null (binary X'00')
- # • shift in (binary X'0E')
- # • shift out (binary X'0F')

- # • space (binary X'40')
- # • plus sign (binary X'4E')
- # • colon (binary X'7A')
- # • equals (binary X'7E')
- # • percent sign (binary X'6C')
- # • backslash (binary X'E0')

If this option is used, the application must ensure that the DELIMITER does not appear in any symbol value in the SYMBOLLIST buffer. For this reason, the application should not use alphanumeric and other printable characters as the DELIMITER value.

DOCSIZE(data-area)

specifies a binary fullword area that will be updated with the current size of the document in bytes. This is the maximum size of the buffer needed to contain a copy of the document when a RETRIEVE command is issued.

DOCTOKEN(data-area)

specifies a data area to contain the symbolic name of the document. The area must be 16 bytes in length and will be set to a CICS-generated name by which the document can be referred to in later commands.

FROM(data-area)

specifies that data supplied by the application is to be used to create the contents of the new document. The data content could be a template or a document which was created and retrieved. If the data is a template,

DOCUMENT CREATE

| symbol substitution will take place where the symbols
| exist in the symbol table. If the data is a previously
| retrieved document, the conversion and bookmark tags
| which were inserted during retrieval will be removed
| from the content and stored in the internal format
| required by the API commands. Note that symbol
| substitution will not be attempted on any unresolved
| symbols contained in a retrieved document.

| FROMDOC(data-area)

| specifies the symbolic name (see the **DOCTOKEN**
| option) of a document whose contents are to be copied
| to the new document being created. All bookmark and
| conversion tags are copied to the new document. The
| symbol table will be not be copied.

| HOSTCODEPAGE(name)

| specifies the name of the host codepage that the data
| being added is encoded in. This option applies to the
| TEXT, SYMBOL and TEMPLATE options only. The
| name must be eight characters long; if it is shorter than
| eight characters it must be padded on the right with
| blanks.

| LENGTH(data-value)

| specifies the length, as a fullword binary value, of the
| application supplied buffer.

| LISTLENGTH(data-value)

| specifies the length, as a fullword binary value, of the
| symbol list.

| SYMBOLLIST(data-area)

| specifies a buffer which contains a symbol list. A symbol
| list is a character string consisting of one or more
| symbol definitions separated by ampersands. Each
| symbol definition consists of a name, an equals sign,
| and a value.

The name is case-sensitive. It may only contain
uppercase and lowercase letters, numbers, and the
special characters dollar ('\$'), underscore ('_'), hyphen
('-'), pound ('#'), period ('.') and at sign ('@'). Unlike the
| symbols in the template, the names in the symbol list
| have neither an ampersand at the beginning, nor a
| semicolon at the end. For example, the symbol &mytitle;
| in the template corresponds to the name *mytitle* in the
| symbol list. Symbols in the symbol list are separated by
| the & character. For example:

| applid=IYCQ&jobname=test

| The following restrictions apply to the characters allowed
| in the value in the symbol list:

- A percent sign may be followed by two hexadecimal digits. When the value is put into the symbol table, the percent sign and the two hexadecimal digits following it are interpreted as the EBCDIC equivalent of the single ASCII character denoted by the two digits. If you want a percent sign in the value in the symbol table, you may put %25 in the symbol list. If the characters following the percent sign are not two valid hexadecimal digits, the

| percent sign and the following characters are put
| into the symbol table as they appear in the symbol
| list. Note that this restriction does not apply if the
UNESCAPED keyword has been specified.

- An ampersand is not allowed. If you want an ampersand in the value in the symbol table, you must put %26 in the value in the symbol list. Note that this restriction does not apply if the DELIMITER keyword is used to specify a delimiter other than ampersand.
- When the value is put into the symbol table, a plus sign is interpreted as a space. If you want a plus sign in the value in the symbol table, you must put %2B in the value in the symbol list. Note that this restriction does not apply if the UNESCAPED keyword has been specified.
- If you want a space in the value in the symbol table, the value in your symbol list may contain a space, a plus sign, or %20. Note that this restriction does not apply if the UNESCAPED keyword has been specified.

| TEMPLATE(name)

| specifies the 48-byte name of a template. The template
| must be defined to CICS using RDO. If the name is
| shorter than 48 bytes, it must be padded on the right
| with blanks.

| TEXT(data-area)

| specifies a buffer of data which is to be used as the
| contents of the new document being created. The data
| is copied unchanged to the document content and no
| attempt is made to parse the data for symbol substitution.
| The data will be marked as requiring conversion to the
| client code page when the document is sent.

UNESCAPED

prevents CICS from unescaping symbol values
contained in the SYMBOLLIST buffer. If this option is
used, plus signs are not converted to spaces, and
sequences such %2B are not converted to single byte
values.

| Conditions

| INVREQ

| RESP2 values are:

- 1 The retrieved document specified on the FROM option is not in a valid RETRIEVE format.

| NOTFND

| RESP2 values:

- 2 The document specified on the FROMDOC option could not be found or was named incorrectly.
- 3 The template specified on the TEMPLATE option could not be found or was named incorrectly.

| **7** The host codepage specified on the
| HOSTCODEPAGE option could not be found or
| was named incorrectly.

8 The value specified for DELIMITER is not valid.

| **SYMBOLERR**

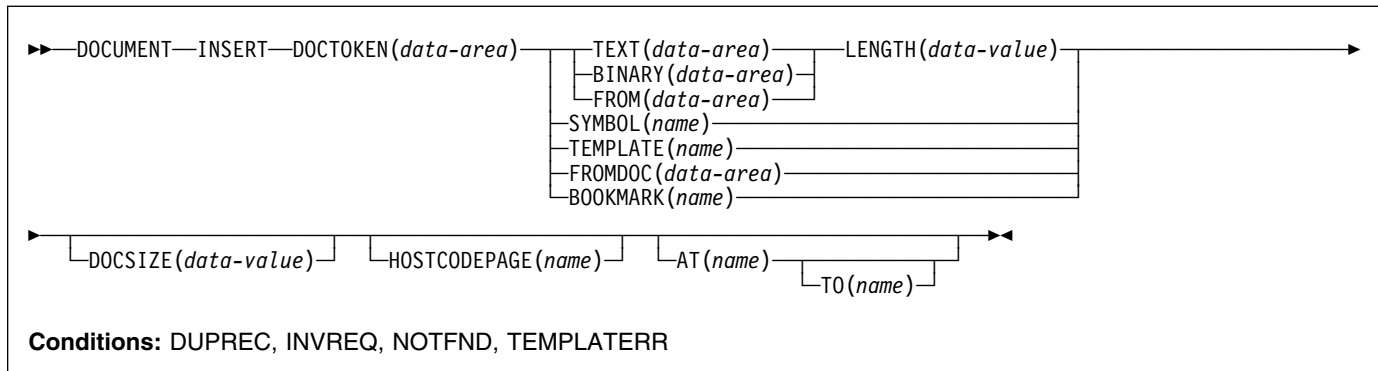
| a symbol specified in the symbol list does not conform to
| the naming rules for symbols. RESP2 contains the offset
| of the symbol in the list.

| **TEMPLATERR**

| an invalid #set, #include or #echo command has been
| encountered while processing the supplied template
| data. RESP2 contains the offset of the invalid command.

DOCUMENT INSERT

Insert document objects.



Description

DOCUMENT INSERT allows the application to insert document objects at insertion points within the document. The insertion points (bookmarks) define relative positions within the document. Bookmarks must be defined before being referenced. Data is always inserted after the position identified by the bookmark.

Options

AT(*name*)

specifies the 16-byte symbolic name of a bookmark which identifies the position of the insertion point in the document. Data is inserted after the bookmark, and any data following the bookmark is shifted down. The application can use a combination of the AT and TO options to perform an overlay operation. If the AT operand is not specified, the data is inserted at the end of the document. A pre-defined bookmark of TOP is provided to allow the application to insert data at the beginning of the document.

BINARY(*data-area*)

specifies a buffer of data to be inserted into the document. The data is copied unchanged to the insertion point in the document, and no attempt is made to parse the data for symbol substitution. The BINARY option allows the application to insert blocks of data that must not undergo conversion to a client code page when the data is sent.

BOOKMARK(*name*)

specifies a bookmark to be inserted into the document. A bookmark is a symbolic name which identifies an insertion point in the document. The name can be up to 16 characters in length, and must not contain any imbedded spaces.

DOCSIZE(*data-value*)

specifies a binary fullword area to be updated with the current size of the document in bytes. This is the maximum size of the buffer needed to contain a copy of the document when a RETRIEVE command is issued.

DOCTOKEN(*data-area*)

specifies the 16-byte symbolic name of the document into which data is to be inserted.

FROM(*data-area*)

specifies that a buffer of data supplied by the application is to be inserted into the document. The data content can be a template or a document that was previously created and retrieved. If the data is a template, symbol substitution takes place where the symbols exist in the symbol table. If the data is a previously retrieved document, the conversion and bookmark tags which were inserted during the retrieval will be removed from the content and stored in the internal form required by the API commands. Note that symbol substitution will not be attempted on any unresolved symbols contained in a retrieved document.

FROMDOC(*data-area*)

specifies the symbolic name of a document (see the DOCTOKEN option) whose contents are copied to the insertion point of the target document. All bookmarks and conversion tags are copied to the target document. The symbol table is not copied.

HOSTCODEPAGE(*name*)

specifies the symbolic name (see the DOCTOKEN option) of the host codepage that the data being added is encoded in. This option applies to the TEXT, SYMBOL and TEMPLATE options only. The name must be eight characters long; if it is shorter than eight characters, it must be padded on the right with blanks.

| LENGTH(data-value)

| specifies the length, as a fullword binary value, of the
| buffer containing the TEXT, BINARY or FROM data.

When the DOCUMENT INSERT command follows a
DOCUMENT RETRIEVE command, without the use of
the DATAONLY option, and the retrieved document is
being inserted using the FROM option, the LENGTH
specified must be equal to the length of the retrieved
document.

| SYMBOL(name)

| specifies the 32-byte name of a symbol in the symbol
| table. The data associated with the symbol in the symbol
| table is inserted, but not the symbol itself.

| TEMPLATE(name)

| specifies the 48-byte name of a template. The template
| must be defined to CICS using RDO. If the name is less
| than 48 bytes, it must be padded on the right with
blanks. The current values of any symbols are substituted
into the template.

| TEXT(data-area)

| specifies a buffer of data to be inserted into the
| document. The data is copied unchanged to the insertion
| point in the document, and no attempt is made to parse
| the data for symbol substitution. When the document is
| sent, it is marked as requiring conversion to the client
| code page.

| TO(name)

| specifies the symbolic name of a bookmark identifying
| the end position of an overlay operation. Data between
| the bookmarks identified by the AT and TO operands is
| deleted, and new data is inserted in its place. It is
| possible to delete data between two bookmarks by
| specifying a null string on the TEXT or BINARY option
| with a LENGTH of zero.

| Conditions**| DUPREC**

| the bookmark has already been defined.

| INVREQ

| RESP2 values are:

| **0** The bookmark specified on the TO option
| appears before the bookmark specified on
| the AT bookmark.

| **1** The retrieved document specified on the
| FROM option is not in a valid RETRIEVE
| format.

| **2** The bookmark name on the BOOKMARK
| option is invalid.

| NOTFND

| one of the following documents or templates could not
| be found, or its name was incorrect.

| RESP2 values:

| **1** The document specified on the DOCUMENT
| option.

| **2** The document specified on the FROMDOC option.

| **3** The template specified on the TEMPLATE option.

| **4** The document specified on the SYMBOL option.

| **5** The document specified on the AT option.

| **6** The document specified on the TO option.

| **7** The document specified on the HOSTCODEPAGE
| option.

| TEMPLATERR

| an invalid #set, #include or #echo command has been
| encountered while processing the supplied template
| data. RESP2 contains either a zero (if the maximum of
| 32 levels of embedded templates is exceeded), or the
| offset of the invalid command.

DOCUMENT RETRIEVE

Copy a document into the application's own buffer.

►► DOCUMENT—RETRIEVE—DOCTOKEN(*data-area*)—INTO(*data-area*)—LENGTH(*data-value*)—MAXLENGTH(*data-value*)—►

► CLNTCODEPAGE(*name*) DATAONLY ◄◄

Conditions: LENGERR, NOTFND

Description

DOCUMENT RETRIEVE allows the application to obtain a copy of the document in its own buffer, which it can then manipulate directly. The document is managed by CICS, and the application does not have direct access to the buffer containing the contents of the document. The document exists only for the duration of the current transaction, so the application must retrieve the document and store it if the document is to exist over transaction boundaries. The retrieved document can be used as a basis for a new document by using the FROM option of the DOCUMENT CREATE command.

When the document is retrieved, CICS inserts tags into the document contents to identify the bookmarks and to delimit the blocks that do not require codepage conversion. To request a copy without tags, specify DATAONLY. The extracted document can also be converted into a single client codepage by using the CLNTCODEPAGE option.

Options

CLNTCODEPAGE(*name*)

specifies the name of the client codepage to which the data should be converted. The name can be up to 40 characters in length; if it is shorter than 40 characters, it must be padded on the right with blanks.

DATAONLY

specifies that the data should be retrieved without any imbedded tags.

DOCTOKEN(*data-area*)

specifies the 16-byte symbolic name of the document to be retrieved.

INTO(*data-area*)

specifies the buffer that is to contain the copy of the document content.

LENGTH(*data-value*)

specifies the length, as a fullword binary value, of the amount of data being returned to the application.

MAXLENGTH(*data-value*)

specifies the length, as a fullword binary value, of the maximum amount of data the buffer can receive.

Conditions

LENGERR

RESP2 values:

- 1 MAXLENGTH is less than or equal to zero. The document is truncated.
- 2 The length of the receiving buffer is zero, or is too short to contain the document contents. The document is truncated.

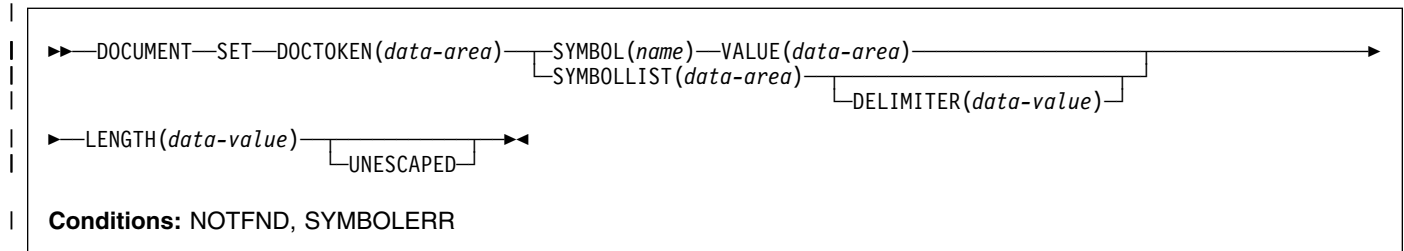
NOTFND

RESP2 values:

- 1 The document has not been created, or the name is incorrectly specified.
- 7 The specified client codepage can not be found.

DOCUMENT SET

Add symbols and values to symbol table.



Description

DOCUMENT SET allows the application to add symbols and their associated values to the symbol table. If the symbol being added already exists in the table, it is replaced by the new definition.

Options

DELIMITER(data-value)

specifies an optional 1-byte value used to delimit symbol name-value pairs in the SYMBOLLIST buffer. If this option is not specified, the value defaults to an ampersand. There are several disallowed DELIMITER values, all of which cause an INVREQ condition if used. The disallowed values are:

- # • null (binary X'00')
- # • shift in (binary X'0E')
- # • shift out (binary X'0F')
- # • space (binary X'40')
- # • plus sign (binary X'4E')
- # • colon (binary X'7A')
- # • equals (binary X'7E')
- # • percent sign (binary X'6C')
- # • backslash (binary X'E0')

If this option is used, the application must ensure that the DELIMITER does not appear in any symbol value in the SYMBOLLIST buffer. For this reason, the application should not use alphanumeric and other printable characters as the DELIMITER value.

| DOCTOKEN(data-area)

| specifies the 16-byte symbolic name of the document that owns the symbol table.

| LENGTH(data-value)

| specifies the length, as a fullword binary value, of the buffer containing the data value associated with the

symbol or the length of the buffer containing the symbol list when the SYMBOLLIST option is used.

| SYMBOL(name)

| specifies the name of the symbol that is to be added to the table. The name can be 1 to 32 characters in length with no embedded spaces.

| SYMBOLLIST(data-area)

| specifies a buffer which contains a symbol list. A symbol list is a character string consisting of one or more symbol definitions separated by ampersands. Each symbol definition consists of a name, an equals sign, and a value.

The name is case-sensitive. It can contain only uppercase and lowercase letters, numbers, and underscores ("_"). Unlike the symbols in the template, the names in the symbol list have neither an ampersand at the beginning, nor a semicolon at the end (for example, the symbol :&mytitle; in the template corresponds to the name *mytitle* in the symbol list). Symbols in the symbol list are separated by the & character. For example:

```
applid=IYCQ&jobname=test
```

The following restrictions apply to the characters allowed in the value in the symbol list:

- A percent sign may be followed by two hexadecimal digits. When the value is put into the symbol table, the percent sign and the two hexadecimal digits following it are interpreted as the EBCDIC equivalent of the single ASCII character denoted by the two digits. If you want a percent sign in the value in the symbol table, you may put %25 in the symbol list. If the characters following the percent sign are not two valid hexadecimal digits, the percent sign and the following characters are put into the symbol table as they appear in the symbol list. Note that this restriction does not apply if the UNESCAPED keyword has been specified.
- An ampersand is not allowed. If you want an ampersand in the value in the symbol table, you must put %26 in the value in the symbol list. Note

DOCUMENT SET

that this restriction does not apply if the DELIMITER
keyword is used to specify a delimiter other than
ampersand.

| • When the value is put into the symbol table, a plus
| sign is interpreted as a space. If you want a plus
| sign in the value in the symbol table, you must put
%2B in the value in the symbol list. Note that this
restriction does not apply if the UNESCAPED
keyword has been specified.

| • If you want a space in the value in the symbol table,
| the value in your symbol list may contain a space, a
plus sign, or %20. Note that this restriction does not
apply if the UNESCAPED keyword has been
specified.

UNESCAPED

prevents CICS from unescaping symbol values
contained in the SYMBOLLIST buffer. If this option is
used, plus signs are not converted to spaces, and
sequences such %2B are not converted to single byte
values.

| VALUE(*data-area*)

| specifies an area containing the value to be associated
| with the SYMBOL.

| Conditions

INVREQ

RESP2 values:

8 The value specified for DELIMITER is not valid.

| NOTFND

| RESP2 values:

| **1** The document has not been created or the name
| is incorrectly specified.

| SYMBOLERR

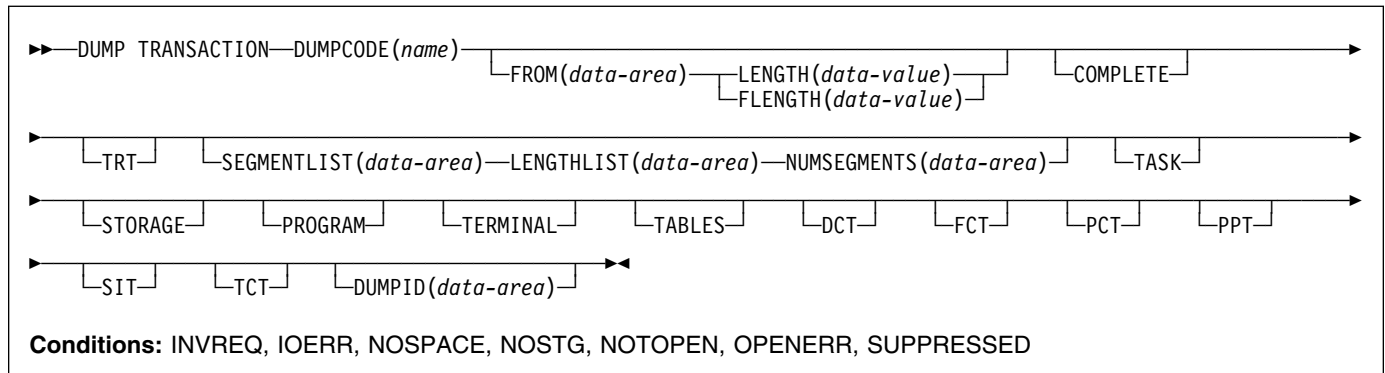
| a symbol name is invalid. RESP2 values:

| **0** SYMBOLLIST was not used.

| **offset** RESP2 contains the offset of the invalid symbol in
| the list.

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.

Note that if you issue a DUMP TRANSACTION for a DUMPCODE that is defined in the transaction dump table with SYSDUMP, you also get a system dump.

If there is no entry in the system dump table for the specified DUMPCODE, a temporary entry is made. This entry is lost on the next CICS start. The system dump table is described in the *CICS Problem Determination Guide*.

Options

COMPLETE

dumps all main storage areas related to a task, all the CICS tables, and the DL/I control blocks.

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, but without the DL/I control blocks.

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 16777215 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 TY SIT parameter, only the trace entries associated with the current task are formatted. If TRTRAN TY=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.

DUMP TRANSACTION

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 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
- DL/I control blocks

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.
- 13** The CICS routine issuing the SDUMP is unable to establish a recovery routine (FESTAE).

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.

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(area1);
storage_length(1)  = CSTG(area1);
:
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 on a local or a remote CICS region.

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 377 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.

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 377 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

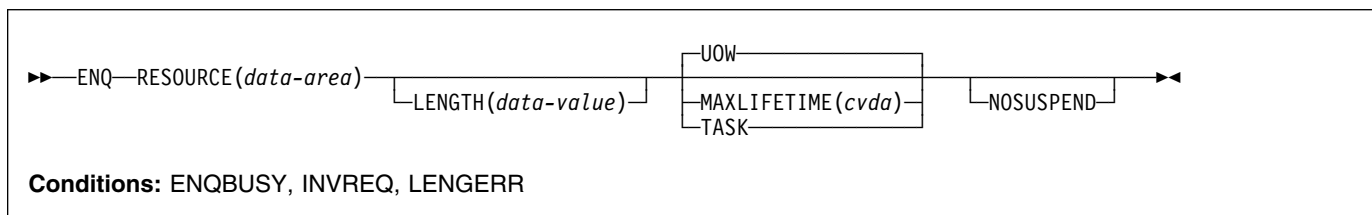
RESP2 values:

- 130** The SYSID option specifies a name that is neither the local system nor a remote system (made known to CICS by defining a 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, unless the name specified in RESOURCE matches a name specified in an installed ENQMODEL resource definition, that has sysplex-wide scope. 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 UOW forces the dequeue at the end of a unit of work (UOW). 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 UOWs.

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. LENGTH is required; the presence of the LENGTH option tells CICS to enqueue on the contents of the data value.

If an ENQMODEL resource definition, CICS checks the value of the ENQSCOPE attribute to determine whether the scope is local or sysplex-wide. If the ENQSCOPE attribute is left blank (the default value), CICS treats the ENQ as local to the issuing CICS region. If no ENQMODEL matches the resource name, the scope of the ENQ command is local. See the *CICS Resource Definition Guide* for more information about the ENQMODEL resource definition.

Resource unavailability: If a resource is not available when ENQ is issued, 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:

When an EXEC CICS ENQ (or DEQ) command is issued for a resource whose name matches that of an installed

UOW

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.

Note: For compatibility with previous releases of CICS/ESA, a CVDA value of LUW is also supported.

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(UOW) 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) UOW
```

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.

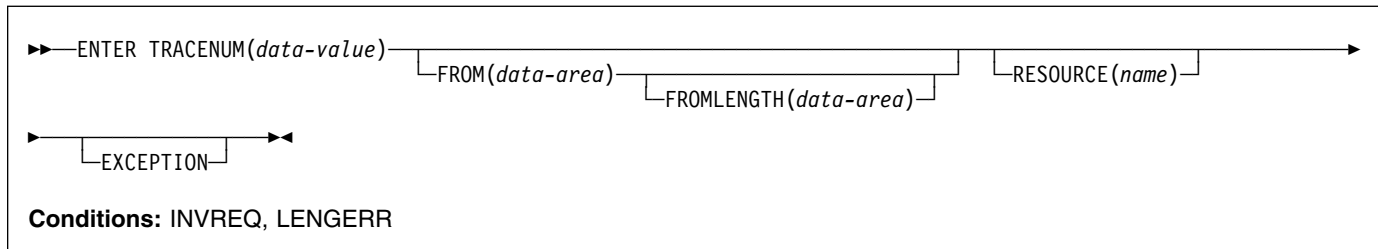
```
EXEC CICS ENQ
      RESOURCE(RESNAME)
```

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 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 Version 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 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.

Default action: terminate the task abnormally.

LENGERR

RESP2 values:

- 4 FROMLENGTH is outside the range 0 through 4000.

Default action: terminate the task abnormally.

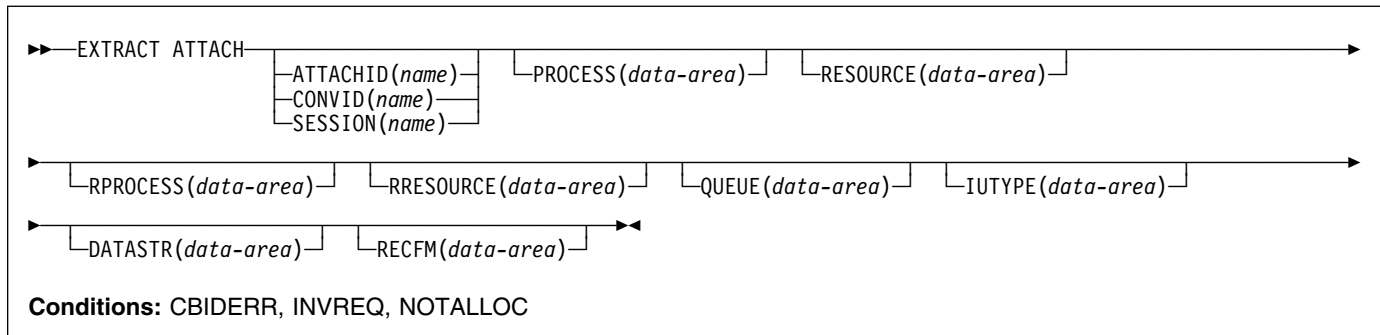
Examples

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 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

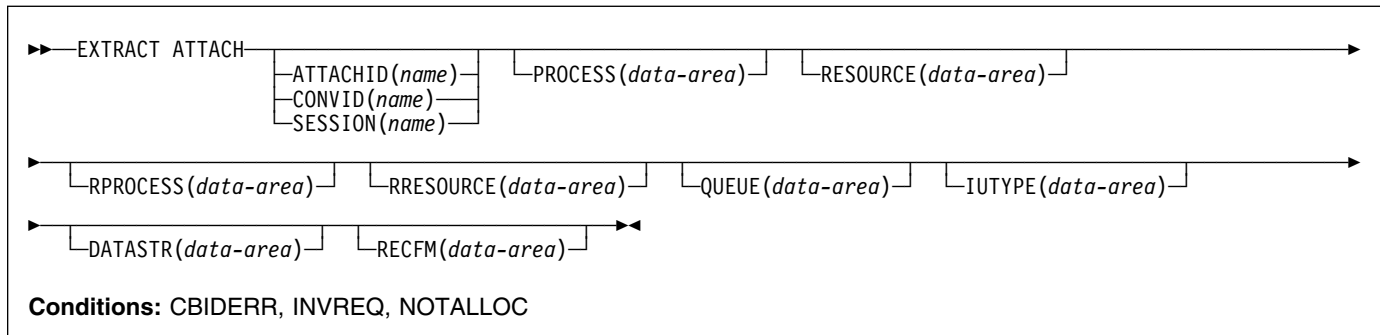
occurs if the facility specified in the command is not owned by the application.

Default action: terminate the task abnormally.

EXTRACT ATTACH (MRO)

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 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—┐—STATE(cvda)—▶▶
                        └─CONVID(name)—┘
```

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.

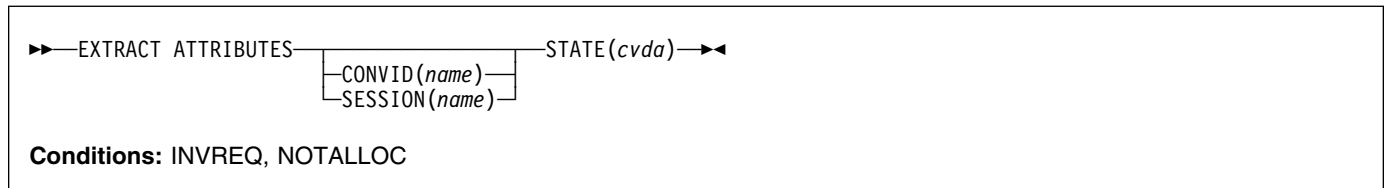
NOTALLOC

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 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.

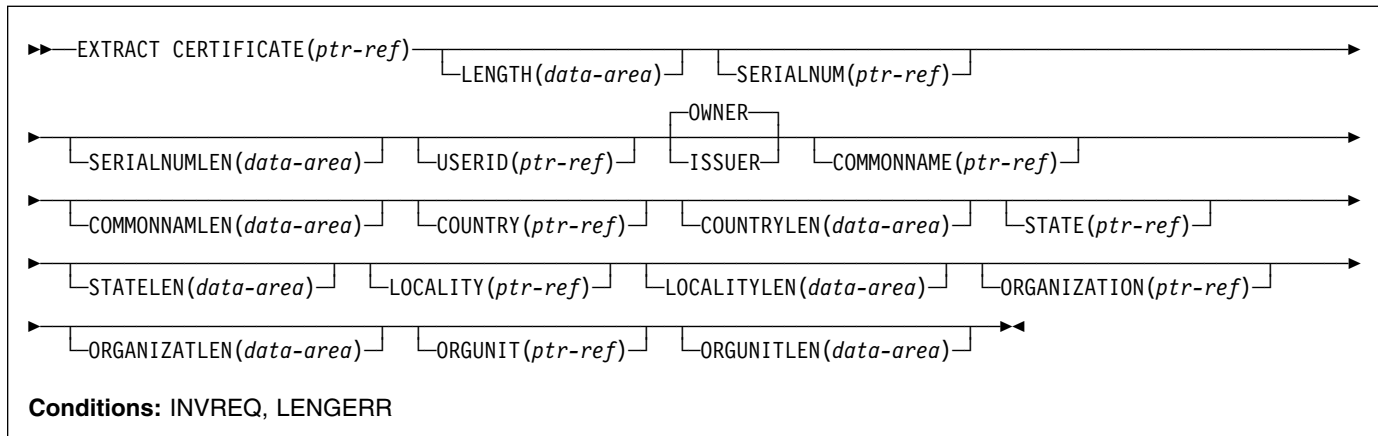
NOTALLOC

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

Default action: terminate the task abnormally.

EXTRACT CERTIFICATE

Obtain information from the client certificate received over a TCP/IP service that specified client authentication.



Description

EXTRACT CERTIFICATE allows the application to obtain information from the X.509 certificate that was received from a client during a Secure Sockets Layer (SSL) handshake over a TCPIP SERVICE that specified SSL(CLIENTAUTH). The certificate contains fields that identify the owner (or subject) of the certificate, and fields that identify the Certificate Authority that issued the certificate. You can select the fields that you require by specifying the OWNER or ISSUER option. You cannot retrieve both OWNER and ISSUER fields with one command.

Options

CERTIFICATE(ptr-ref)

specifies a pointer reference to be set to the address of the full binary certificate received from the client. The pointer reference is valid until the next CICS command or the end of task.

COMMONNAME(ptr-ref)

specifies a pointer reference to be set to the common name from the client certificate. The pointer reference is valid until the next CICS command or the end of task.

COMMONNAMLEN(data-area)

specifies a fullword binary data area to be set to the length of the common name from the client certificate.

COUNTRY(ptr-ref)

specifies a pointer reference to be set to the address of the country from the client certificate. The pointer reference is valid until the next CICS command or the end of task.

COUNTRYLEN(data-area)

specifies a halfword binary data area to be set to the length of the country from the client certificate.

ISSUER

indicates that the values returned by this command refer to the Certificate Authority that issued this certificate.

LENGTH(data-area)

specifies a fullword binary data area to be set to the length of the body of the client certificate.

LOCALITY(ptr-ref)

specifies a pointer reference to be set to the address of the locality from the client certificate. The pointer reference is valid until the next CICS command or the end of task.

LOCALITYLEN(data-area)

specifies a halfword binary data area to be set to the length of the locality from the client certificate.

ORGANIZATION(ptr-ref)

specifies a pointer reference to be set to the address of the organization from the client certificate. The pointer reference is valid until the next CICS command or the end of task.

ORGANIZATLEN(data-area)

specifies a halfword binary data area to be set to the length of the organization from the client certificate.

ORGUNIT(ptr-ref)

specifies a pointer reference to be set to the address of the organization unit from the client certificate. The pointer reference is valid until the next CICS command or the end of task.

| ORGUNITLEN(data-area)

| specifies a halfword binary data area to be set to the
| length of the organization unit from the client certificate.

| OWNER

| indicates that the values returned by this command refer
| to the owner of the certificate.

| SERIALNUM(ptr-ref)

| specifies a pointer reference to be set to the address of
| the serial number of the certificate assigned by the
| certificate issuer. The pointer reference is valid until the
| next CICS command or the end of task.

| SERIALNUMLEN(data-area)

| specifies a halfword binary data area to be set to the
| length of the serial number.

| STATE(ptr-ref)

| specifies a pointer reference to be set to the address of
| the state or province from the client certificate. The
| pointer reference is valid until the next CICS command
| or the end of task.

| STATELEN(data-area)

| specifies a halfword binary data area to be set to the
| length of the state or province from the client certificate.

| USERID(ptr-ref)

| specifies a pointer reference to be set to the userid
| connected with the client certificate. The pointer
| reference is valid until the next CICS command or the
| end of task.

| Conditions**| INVREQ**

| occurs for the following conditions:

- | • the command is being issued in a non-CICS Web
| Interface application.
- | • the command is being issued for a non-HTTP
| request.
- | • if an error occurs retrieving the certificate data from
| CICS intermediate storage.

| LENGERR

| the string being extracted is longer than the length
| specified for one of the options.

EXTRACT LOGONMSG

Access VTAM logon data.

```
▶▶EXTRACT LOGONMSG [INTO(data-area) | SET(ptr-ref)] LENGTH(data-area)▶▶
```

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 for this option.

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 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.

Conditions

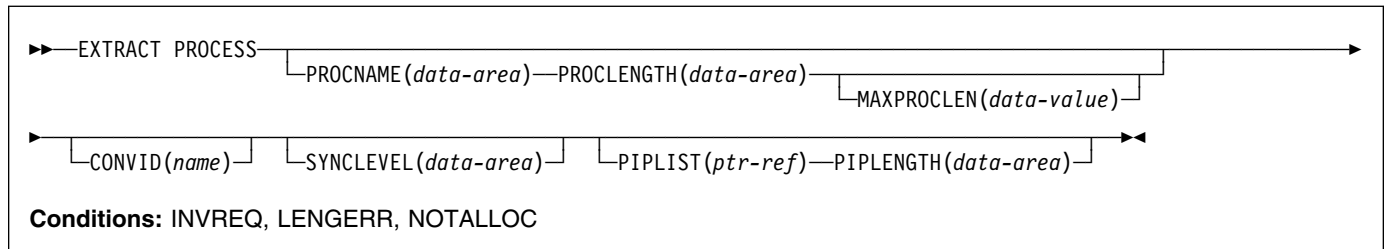
NOTALLOC

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.

PIPLENGTH(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.
- The command is issued against a CPI-Communications conversation.

EXTRACT PROCESS

Default action: terminate the task abnormally.

LENGERR

occurs if the actual length of PROCNAME is greater than MAXPROCLen, or greater than 32 bytes if MAXPROCLen is not specified.

Default action: terminate the task abnormally.

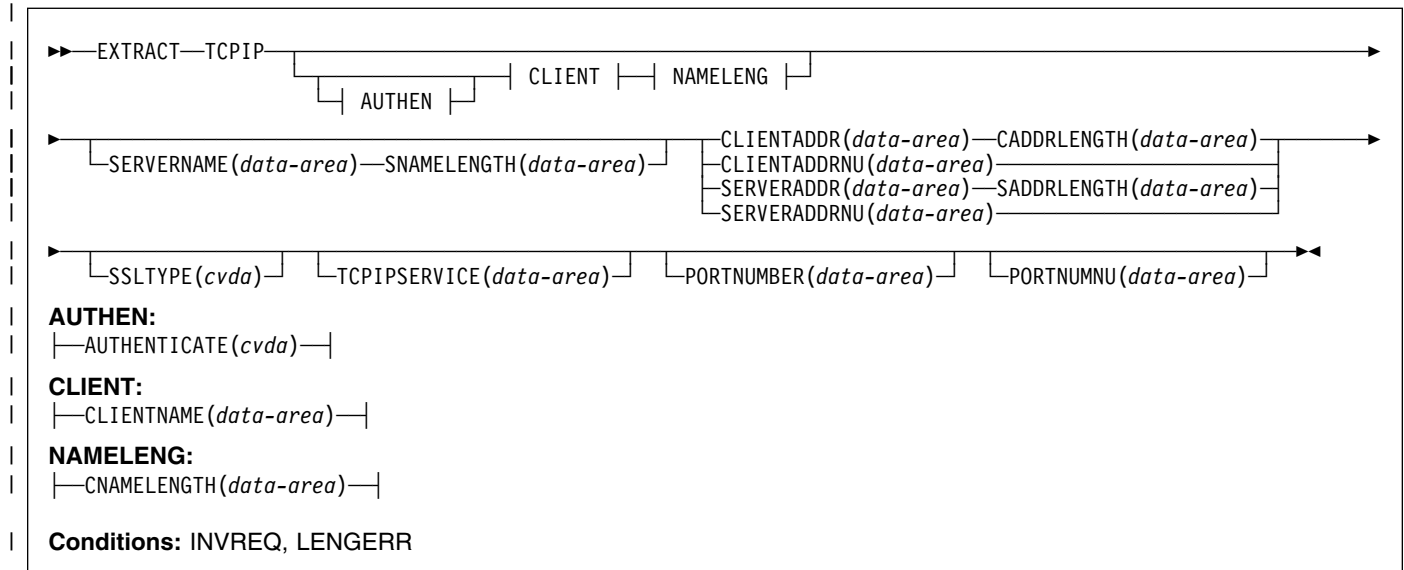
NOTALLOC

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

Default action: terminate the task abnormally.

EXTRACT TCPIP

Obtain information about TCPIP characteristics of the current transaction.



Options

AUTHENTICATE(*cvda*)

returns a CVDA indicating the authentication requested for the client using this transaction. Possible values are:

- # • AUTOAUTH
- # • AUTOREGISTER
- # • BASICAUTH
- # • CERTIFICAUTH
- # • NOAUTHENTIC

| CADDRLENGTH(*data-area*)

| specifies the length of the buffer supplied on the CLIENTADDR option, and is set to the actual length of the data returned to the application. If the data exceeds the buffer length, a LENGERR condition is raised and the data is truncated.

| CLIENTADDR(*data-area*)

| specifies a buffer to contain the client's TCP/IP address.

| CLIENTADDRNU(*data-area*)

| specifies a fullword binary field containing the client's TCP/IP address in binary form.

| CLIENTNAME(*data-area*)

| specifies a buffer to contain the client's name as known by the Domain Name Server.

| CNAMELENGTH(*data-area*)

| specifies the length of the buffer supplied on the CLIENTNAME option, and is set to the actual length of the data returned to the application. If the data exceeds

the buffer length, a LENGERR condition is raised and the data is truncated.

| PORTNUMBER(*data-area*)

| Specifies a 5-character field to contain the port number associated with this transaction in character form. This is the port on which the incoming data that initiated this transaction was received.

| PORTNUMNU(*data-area*)

| Fullword field to contain the port number associated with this transaction in binary form. This is the port on which the incoming data that initiated this transaction was received.

| SADDRLENGTH(*data-area*)

| specifies the length of the buffer supplied on the SERVERADDR option, and is set to the actual length of the data returned to the application. If the data exceeds the buffer length, a LENGERR condition is raised and the data is truncated.

| SERVERADDR(*data-area*)

| specifies a buffer to contain the server's TCP/IP address in dotted decimal character form (nnn.nnn.nnn.nnn).

| SERVERADDRNU(*data-area*)

| specifies a fullword binary field containing the server's TCP/IP address in binary form.

| SERVERNAME(*data-area*)

| specifies a buffer to contain the server's name as known by the Domain Name Server.

| SNAMELENGTH(*data-area*)

| specifies the length of the buffer supplied on the SERVERNAME option, and is set to the actual length of

EXTRACT TCPIP

| the data returned to the application. If the data exceeds
| the buffer length, a LENGERR condition is raised and
| the data is truncated.

| SSLTYPE(CVDA)

| Returns a CVDA indicating whether the Secure Sockets
| Layer (SSL) is being used to secure communications for
| this transaction. Possible values are:

- | • SSL
- | • NOSSL
- | • CLIENTAUTH

| TCPIPSERVICE(data-area)

| An 8-byte field to contain the name of the
| TCPIPSERVICE associated with this transaction.

| Conditions

| INVREQ

RESP2 values:

- # 2 An invalid socket response was received.
- # 5 The command was issued from a non-TCPIP
application.

| LENGERR

RESP2 values:

- # 3 CLIENTADDR is too small to contain the string
extracted.
- # 4 SERVERADDR is too small to contain the string
extracted.
- # 6 CLIENTNAME is too small to contain the string
extracted.
- # 7 SERVERNAME is too small to contain the string
extracted.

EXTRACT TCT

Convert an 8-character name to a 4-character name on an LUTYPE6.1 logical unit.

```
▶▶—EXTRACT TCT—NETNAME(name)—
      ┌—SYSID(systemname)—
      └—TERMIN(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.

TERMIN(*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.

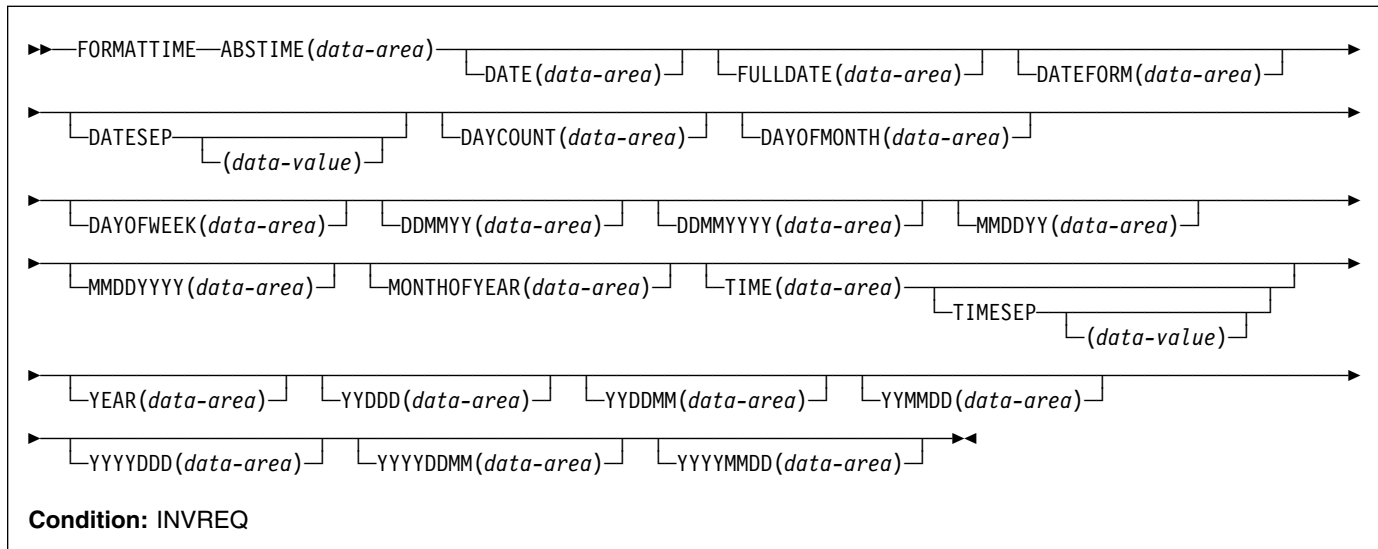
NOTALLOC

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.

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.

FREE

Conditions

INVREQ

RESP2 values:

- # 1 The ABSTIME value is less than zero or not in packed-decimal format.
- #

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

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 EDF, and the transaction frees the principal facility, EDF 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 EDF, and the transaction frees the principal facility, EDF 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

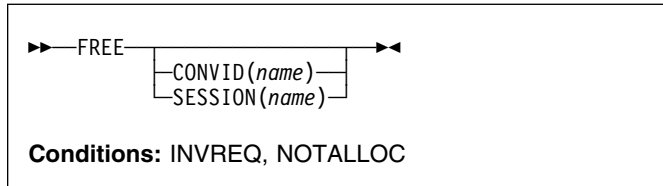
occurs if the specified CONVID value does not relate to a conversation owned by the application.

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 EDF, and the transaction frees the principal facility, EDF 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 377.)

Default action: terminate the task abnormally.

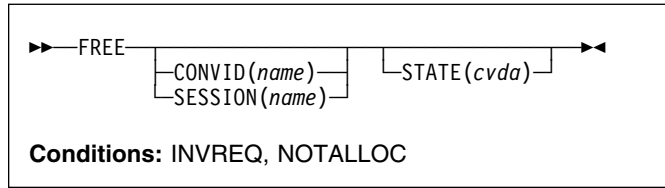
NOTALLOC

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 EDF, and the transaction frees the principal facility, EDF 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 377.)

Default action: terminate the task abnormally.

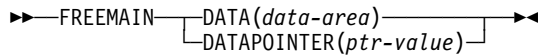
NOTALLOC

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.



Condition: INVREQ

Note for dynamic transaction routing

Using FREEMAIN of storage GETMAINed with SHARED, or of a resource defined with 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, map, or table, defined with RELOAD=YES. 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:

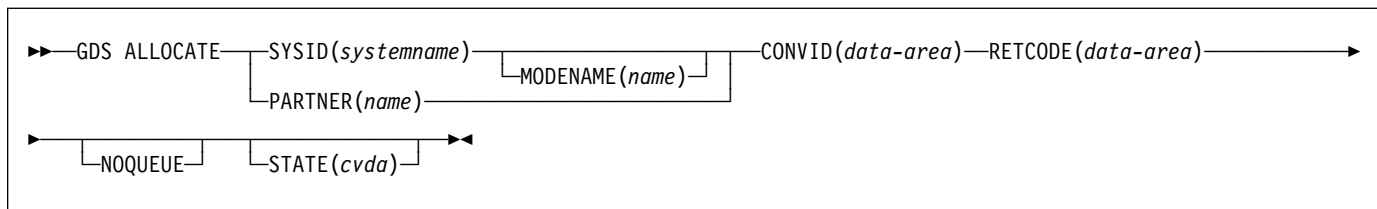
```
WORKAREA DS CL100
.
.
EXEC CICS GETMAIN SET(9) LENGTH(100)
USING WORKAREA,9
EXEC CICS FREEMAIN DATA(WORKAREA)
```

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 1 on page 119). 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*)

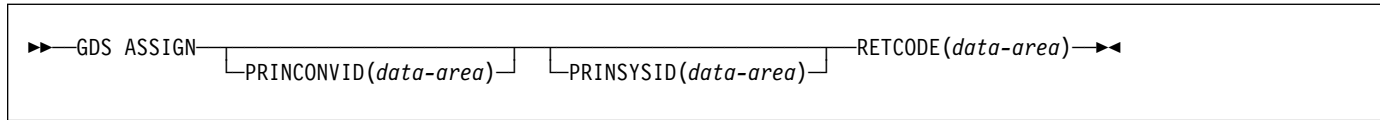
specifies the remote system to which an APPC session is to be allocated. The name, which is 1–4 characters, identifies an entry (defined as an APPC connection) in the CICS terminal control table.

Table 1. GDS ALLOCATE return codes

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 2). 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.

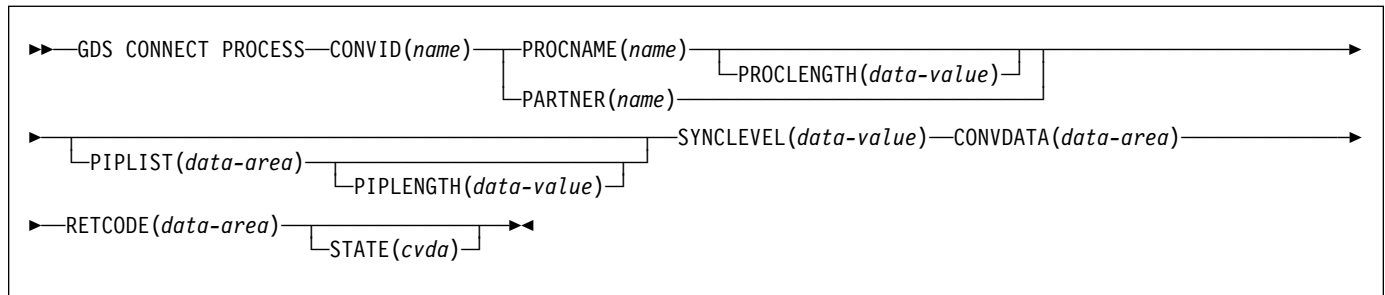
RETCODE(*data-area*)

specifies the 6-byte application data area into which return code information is to be moved.

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 3 on page 122). 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.

PIPLIST(data-area)

specifies the PIP data that is to be sent to the remote process. See the *CICS Distributed Transaction Programming 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. You can also use the TPNAME value in the transaction 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
  
```

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 PIPLNGTH 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 PIPLNGTH 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 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.

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:

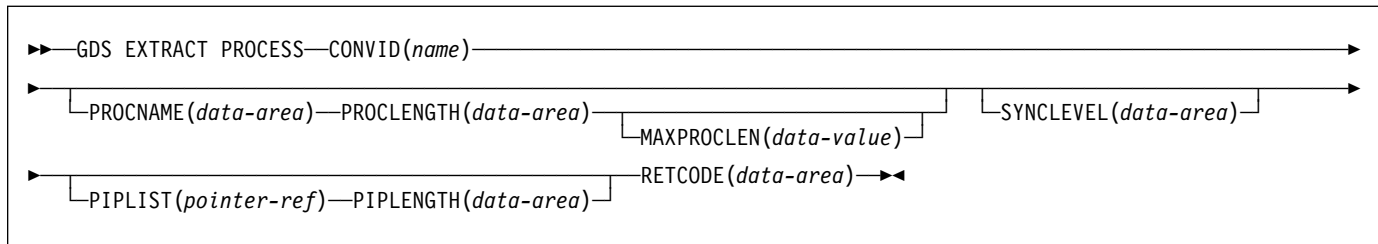
ALLOCATED
 CONFFREE
 CONFRECEIVE
 CONFSEND
 FREE
 PENDFREE
 PENDRECEIVE
 RECEIVE
 ROLLBACK
 SEND
 SYNCFREE
 SYNCRECEIVE
 SYNCSEND

Table 4. 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 5). 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.

PIPELENGTH(*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 5. 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 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.

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
SYNCSEND

Table 6. 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 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.

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 7. 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 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 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 8. 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).

```
▶▶—GDS ISSUE ERROR—CONVID(name)—CONVDATA(data-area)—RETCODE(data-area)—└─STATE(cvda)—◀◀
```

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
 SYNCSEND

Table 9. 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 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
 CONFREE
 CONFRECEIVE
 CONFSEND
 FREE
 PENDFREE
 PENDRECEIVE
 RECEIVE
 ROLLBACK
 SEND
 SYNCFREE
 SYNCRECEIVE
 SYNCSEND

Table 10. 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 11). 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:

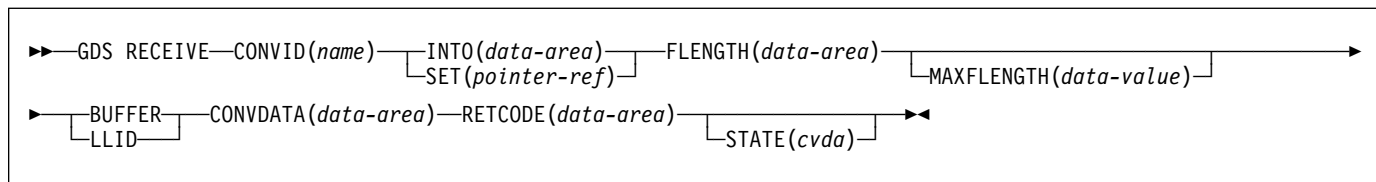
ALLOCATED
 CONFREE
 CONFRECEIVE
 CONFSEND
 FREE
 PENDFREE
 PENDRECEIVE
 RECEIVE
 ROLLBACK
 SEND
 SYNCFREE
 SYNCRECEIVE
 SYNCSEND

Table 11. 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 12 on page 132). 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*)

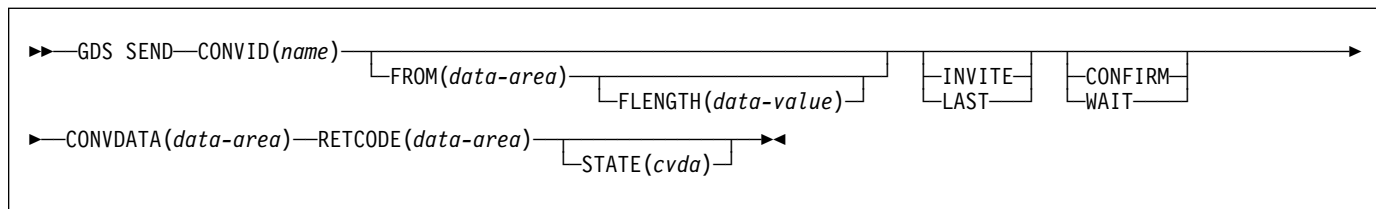
gets the state of the current conversation. The *cvda* values returned by CICS are:

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 13 on page 134). 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 1–32767) 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
 RECEIVE
 ROLLBACK
 SEND
 SYNCFREE
 SYNCRECEIVE
 SYNCSEND

GDS SEND

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 14). 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:

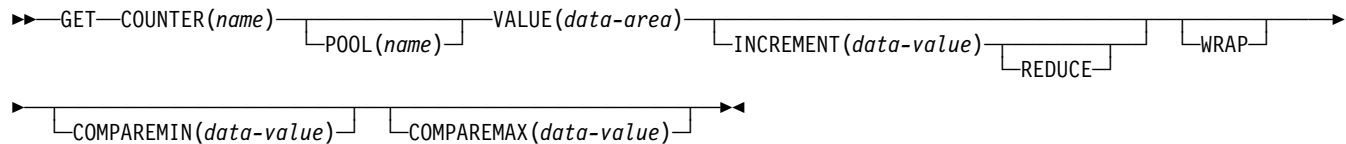
ALLOCATED
CONFFREE
CONFRECEIVE
CONFSEND
FREE
PENDFREE
PENDRECEIVE
RECEIVE
ROLLBACK
SEND
SYNCFREE
SYNCRECEIVE
SYNCSEND

Table 14. GDS WAIT return codes

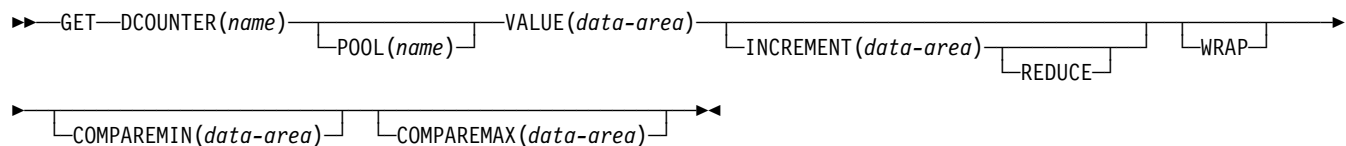
RETCODE (hexadecimal)	Description
03 00	CONVID is for a session that is not APPC.
03 00	CONVID is for a session that is in use by CPI Communications.
03 04	CONVID is for a conversation that is not basic.
03 08	A state check has occurred.
04	CONVID is for a session that is not allocated to task, or that is a relay link.

GET COUNTER

Get the next number from the named counter.



Conditions: INVREQ, LENGERR, SUPPRESSED



Conditions: INVREQ, SUPPRESSED

Description

These counter commands obtain, from the named counter server, the current number from the named counter in the specified pool, and updates the current number by the default, or by a specified, increment. The default increment is 1. COUNTER operates with fullword *signed* binary values, and DCOUNTER operates with doubleword *unsigned* binary values.

You can use the COMPAREMAX and COMPAREMIN options to obtain a number only if it falls within a specified range, or is above or below a specified value.

For information about specifying fullword and doubleword variables on these named counter commands, see “Argument values” on page 2.

Options

COMPAREMAX(*data-value*)

specifies, as a fullword signed binary value (doubleword unsigned binary value for DCOUNTER), a value to be compared with the named counter's current value, and makes the result of the GET command conditional on the comparison:

- If the current value to be assigned is less than, or equal to, the value specified on the COMPAREMAX parameter, the current value is returned, with response normal
- If the current value is greater than the specified value, CICS returns an exception condition.

The value you specify on the COMPAREMAX parameter can be less than the value on the COMPAREMIN parameter, in which case the current value is considered to be in range if it satisfies the COMPAREMIN or the COMPAREMAX comparison. In the normal case where the COMPAREMIN value is less than the COMPAREMAX value, the current value must satisfy both comparisons (that is, it must be greater than, or equal to, the COMPAREMIN value, and be less than, or equal to the COMPAREMAX value).

COMPAREMIN(*data-value*)

specifies, as a fullword signed binary value (doubleword unsigned binary value for DCOUNTER), a value to be compared with the named counter's current value, and makes the result of the GET command conditional on the comparison:

- If the current value to be assigned is equal to, or greater than, the value specified on the COMPAREMIN parameter, the CICS returns the current value, with response normal
- If the current value is less than the specified value, CICS returns an exception condition.

Note: The value you specify on the COMPAREMIN parameter can be greater than the value on the COMPAREMAX parameter. See the COMPAREMAX parameter for the effect of this.

COUNTER(*name*)

specifies the name of the fullword counter from which the current number is to be assigned to the application program. The name can be up to 16 alphanumeric characters. If *name* is a variable that contains a name

that is less than 16 characters, the name must be padded with trailing blanks.

DCOUNTER(*name*)

specifies the name of the doubleword counter from which the current number is to be assigned to the application program. The name can be up to 16 alphanumeric characters. If *name* is a variable that contains a name that is less than 16 characters, the name must be padded with trailing blanks.

INCREMENT(*data-value*)

specifies, as a fullword signed binary value (doubleword unsigned binary value for DCOUNTER), an increment by which the named counter is to be updated, instead of the default value of 1. The counter is incremented after the current number has been assigned.

Specifying an increment to override the default increment of 1 enables the application program to obtain exclusive use of more than 1 number for each call. For example, if you want to obtain exclusive use of a block of 20 numbers, specify INCREMENT(20).

See the description of the REDUCE and WRAP options for the effect of specifying an increment when the counter is at, or near, the maximum value.

POOL(*poolname*)

specifies an 8-character string to be used as a pool selection parameter to select the pool in which the named counter resides. The string can be a logical pool name, or the actual pool name.

Valid characters for the pool selector string are A through Z, 0 through 9, \$ @ # and _ (underscore). If *name* is a variable that contains a name that is less than 8 characters, the name must be padded with trailing blanks.

This parameter is optional. If you omit the name of the pool, a pool selector value of 8 blanks is assumed.

If there is no matching entry in the DFHNCOPT options table, CICS uses the name specified on the NCPLDFT system initialization parameter, which specifies the default named counter pool.

For information about generating a named counter options table using the DFHNCO macro, see the *CICS System Definition Guide*.

REDUCE

specifies that you want the named counter server to reduce the specified increment if the range of numbers remaining to be assigned is too small.

The range of numbers is too small if the difference between the current value and the maximum value plus 1 is less than the specified increment, in which case:

- If you specify REDUCE, the INCREMENT parameter value is reduced and the GET request succeeds. In this case, the GET command has reserved a range of numbers less than that specified by the INCREMENT parameter, and the

current value is updated to the maximum value plus 1.

- If you do not specify the REDUCE option, the result depends on whether or not you specify the WRAP option. If the REDUCE and WRAP options are both omitted, the request fails with the counter-at-limit error (SUPPRESSED, RESP2=101), but the current number is not changed. For example, if a request specifies an INCREMENT parameter value of 15 when the current number is 199 990 and the counter maximum number is defined as 199 999, the GET command fails because updating the counter by the specified increment would cause the current number to exceed 200 000.

See the WRAP parameter description for information about the effect of WRAP when REDUCE is omitted.

VALUE(*data-area*)

specifies the data area (fullword signed data-area for COUNTER, and doubleword unsigned data-area for DCOUNTER) into which CICS returns the current number, obtained from the named counter server for the specified pool.

WRAP

specifies that you want the named counter server automatically to perform a rewind of the named counter if it is in a counter-at-limit condition, avoiding the error condition that would otherwise result.

If it finds that the named counter is in the counter-at-limit condition, or that the increment specified without the REDUCE option would cause the condition, the server:

- Resets the current value of the named counter equal to the minimum value defined for the counter
- Returns the new current value to the application program, with DFHRESP(NORMAL)
- Updates the current value by the required increment ready for the next request.

If you omit the WRAP option, and the counter-at-limit condition has been reached, CICS returns SUPPRESSED, RESP2=101.

Conditions

INVREQ

RESP2 values:

- 201** Named counter not found.
- 301** The server has reported an error code that is not understood by the named counter interface. Generally, this is not possible unless the interface load module, DFHNCF, is at a lower maintenance or release level than the server itself.
- 303** An unexpected error, such as structure failure or loss of connectivity, has occurred on a macro used to access the coupling facility. Further

GET COUNTER

| information can be found in message DFHNC0441
| in the CICS job log.

| **304** The pool selection parameter specified in the
| program cannot be resolved to a valid server
| name using the current options table.

| **305** The interface is unable to establish a connection
| to the server for the selected named counter pool.
| Further information can be found in an AXM
| services message (AXMSCnnnn) in the CICS job
| log.

| **306** An abend occurred during server processing of a
| request. Further information can be found in a
| message in the CICS job log and the server job
| log.

| **308** The DFHNCOPT options table module, required
| for resolving a pool name, could not be loaded.

| **309** During processing of the options table, the named
| counter interface encountered an unknown entry
| format. Either the options table is not correctly
| generated, or the DFHNCIF interface load module
| is not at the same release level as the options
| table.

| **310** An options table entry matching the given pool
| name specified a user exit program, but the user
| exit program is not link-edited with the options
| table and cannot be loaded.

| **403** The POOL parameter contains invalid characters
| or embedded spaces.

| **404** The COUNTER parameter contains invalid
| characters or embedded spaces.

| **406** The INCREMENT value is invalid. The value
| specified cannot be greater than the total range of
| the counter ((maximum value – minimum value)
| + 1).

| Default action: terminate the task abnormally.

LENGERR

| LENGERR occurs for COUNTER commands only and
| does not apply to DCOUNTER requests. It occurs when
| a counter that was defined by a DCOUNTER command
| has become negative when accessed as a fullword
| signed binary value (that is, the counter uses more than
| 31 bits).

| In each of the three cases of overflow, the named
| counter server completes the operation, and returns a
| warning response to CICS, which CICS returns to your
| application program as the RESP2 value. The data area
| contains the low-order 32 bits returned from the named
| counter server, which could be a negative number.

| RESP2 values:

| **001** The current value that the server has attempted to
| return in the VALUE data area has overflowed into
| the high-order (sign) bit (that is, the value returned
| is negative).

| **002** The current value is too large for a fullword data
| area by only 1 bit. In this case, the overflow value
| is exactly 1.

| **003** The current value is too large for a fullword data
| area by a value greater than 1.

| Default action: terminate the task abnormally.

SUPPRESSED

| RESP2 values:

| **101** The maximum value for the named counter has
| already been assigned and the counter is in the
| 'counter-at-limit' condition. No more counter
| numbers can be assigned until the named counter
| has been reset, either by a REWIND command, or
| by specifying the WRAP option on the GET
| command.

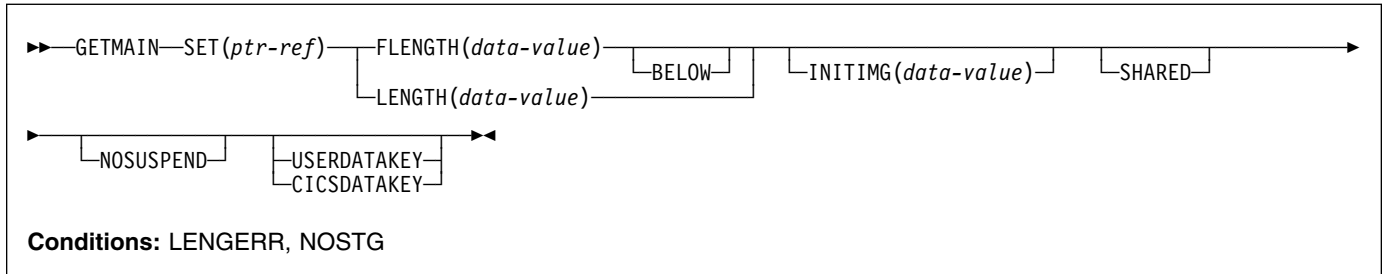
| **103** The current value of the named counter is:

- not within the range specified by the COMPAREMAX and COMPAREMIN parameters, when both are specified
- greater than the COMPAREMAX parameter or less than the COMPAREMIN parameter, when only one option is specified.

| Default action: terminate the task abnormally.

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 under which the requesting program is running if the USERDATAKEY or CICSDATAKEY option is 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:

GETMAIN

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, including those that are running with transaction isolation.

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 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 transaction resource definition.

FLENGTH(*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.

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 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 transaction resource definition.

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 139.

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 139.

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)
```

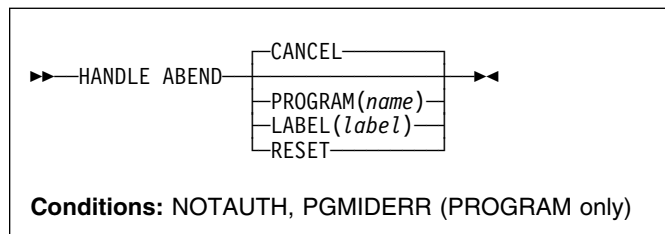
You must define BLANK in your program as the character representing a space.

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 transaction resource definition), and initialize it to spaces:

```
EXEC CICS GETMAIN SET(PTR)
          FLENGTH(2048)
          INITIMG(BLANK)
          CICSDATAKEY
```

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 and execution key used are those of the program that issued the HANDLE ABEND command.

If PROGRAM is specified, the addressing mode is defined by the way the program is link-edited and the execution key is specified by the EXECKEY option on the program's resource definition.

If a COMMAREA has been established, it will be passed to the specified PROGRAM.

HANDLE AID

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.

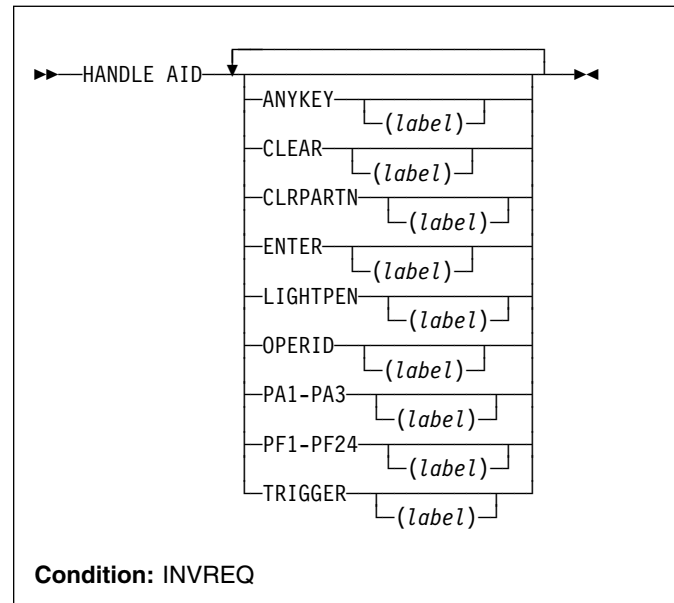
Examples

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 411.

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.

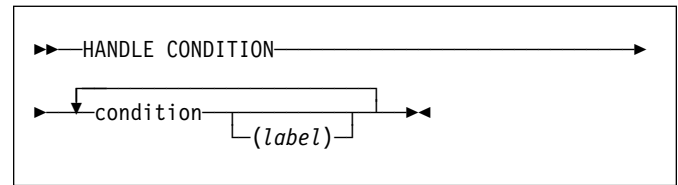
Examples

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.

IGNORE CONDITION

- 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. The addressing mode is set to that in effect at the point where the HANDLE CONDITION command is issued.

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 377.

Examples

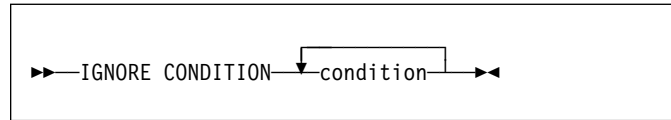
The following example shows you how to handle conditions, such as DUPREC, LENGERR, and so on, that can occur when you use a WRITE command to add a record to a data set.

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 conditions.



Description

IGNORE CONDITION is not supported for C programs.

For information about the conditions, see Appendix A, “EXEC interface block” on page 377.

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.

ISSUE ABEND

Abend the mapped conversation with an APPC partner.

```

▶▶—ISSUE ABEND—┐┐CONVID(name)┐┐STATE(cvda)┐┐◀◀

```

Conditions: INVREQ, NOTALLOC, TERMERR

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
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:

- 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

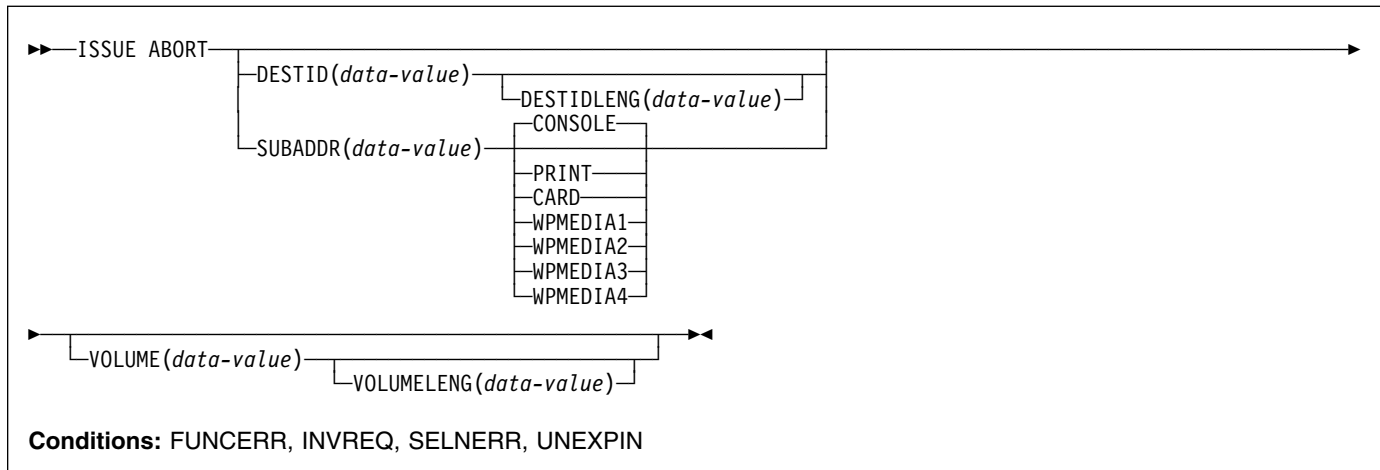
occurs for a session-related error. Any action on that conversation other than a FREE command causes an ATCV abend.

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.

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.

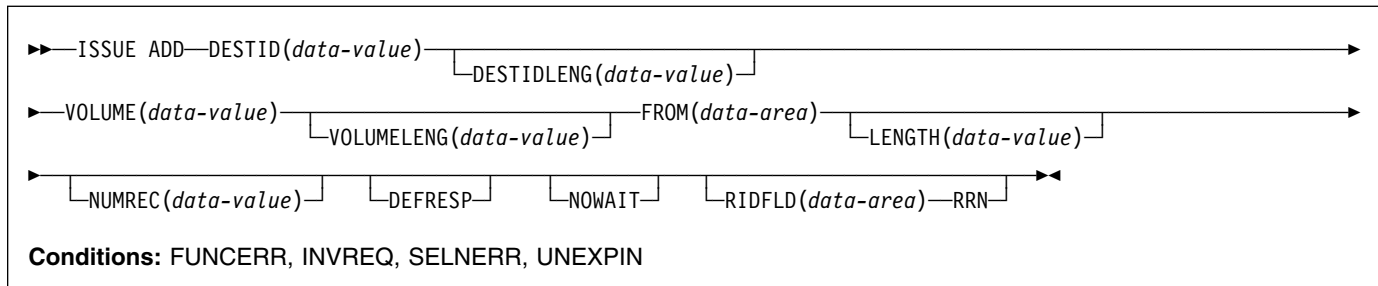
UNEXPIN

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.

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.

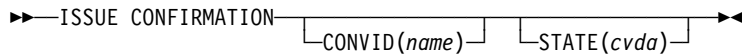
UNEXPIN

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.



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).

```
▶▶—ISSUE COPY—TERMINID(name)—┌CTLCHAR(data-value)┐ ┌WAIT┐ ▶▶
```

Condition: NOTALLOC, TERMIDERR

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 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).

Default action: terminate the task abnormally.

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)

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 information display system (BTAM or TCAM)
- 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

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.

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.

ISSUE DISCONNECT (LUTYPE6.1)

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 set in the 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

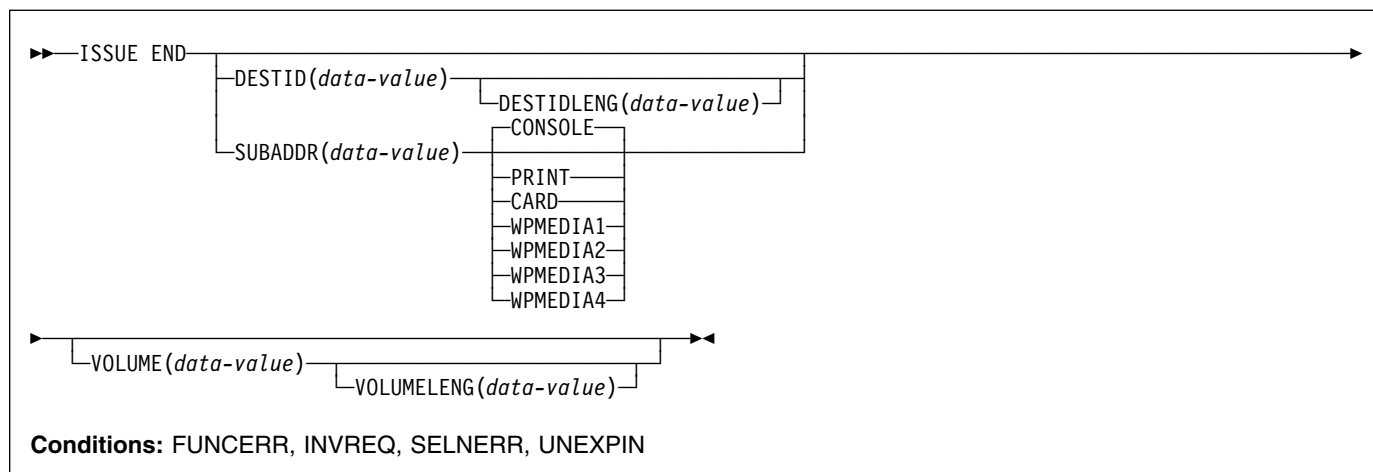
occurs for a terminal-related error, such as a session failure.

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.

ISSUE ENDFILE

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.

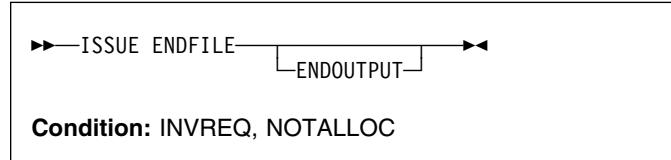
UNEXPIN

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.



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.

```
▶▶—ISSUE ENDOUTPUT—┐
                    └─ENDFILE─┘▶▶
```

Condition: INVREQ, NOTALLOC

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

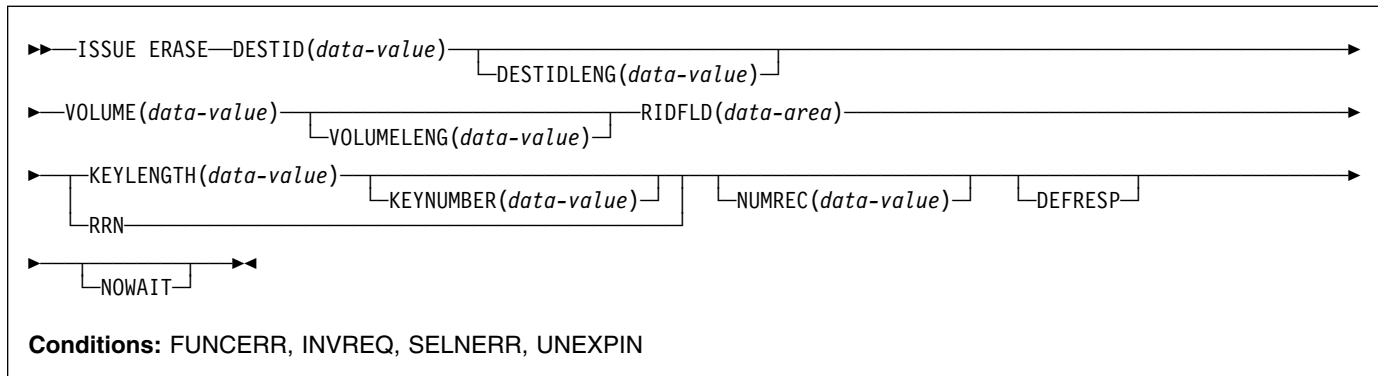
occurs for a terminal-related error, such as a session failure.

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.

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.

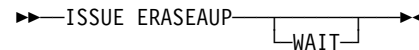
UNEXPIN

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 information display system (BTAM or TCAM)
- 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.

ISSUE ERASEAUP

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.

```

▶—ISSUE ERROR—┬─CONVID(name)─┬─STATE(cvda)─▶

```

Conditions: INVREQ, NOTALLOC, SIGNAL, TERMERR

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

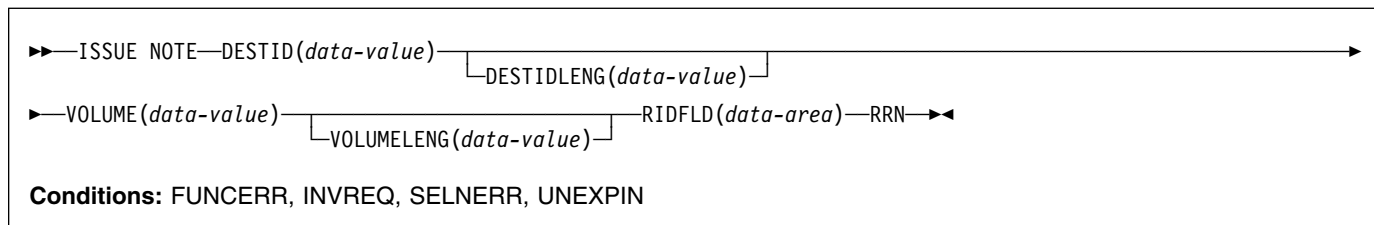
occurs if the facility specified in the command is not owned by the application.
Default action: terminate the task abnormally.

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.

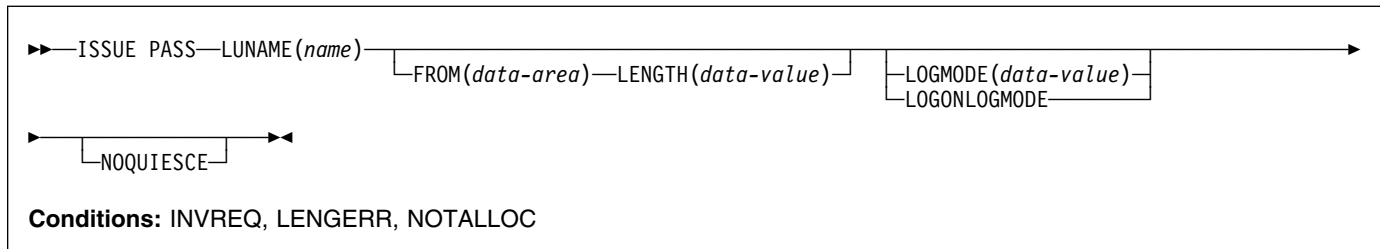
UNEXPIN

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 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 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 (PSDINT=nnn in the SIT) is in use, then the TYPETERM 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.

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
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 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 on the RDO TERMINAL resource definition, or by a printer supplied by the autoinstall user program.
- For a 3270-display logical unit with the PTRADAPT feature, used with a 3274 or 3276, the printer is allocated by the printer authorization matrix. The PTRADAPT 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.

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

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.

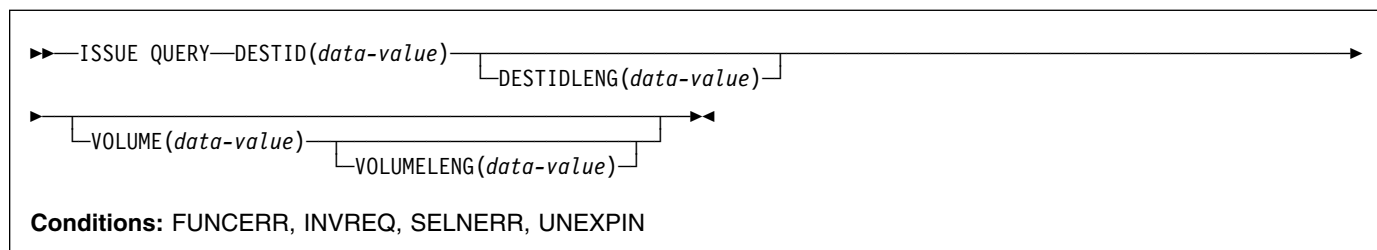
TERMERR

occurs for a terminal-related error.

An ISSUE PRINT on a BTAM 3270 information display system cannot raise the TERMERR condition.

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.

```

ISSUE RECEIVE [INTO(data-area)] [SET(ptr-ref)] [LENGTH(data-area)]
  
```

Conditions: DSSTAT, EOC, EODS, INVREQ, LENGERR, UNEXPIN

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 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.

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.

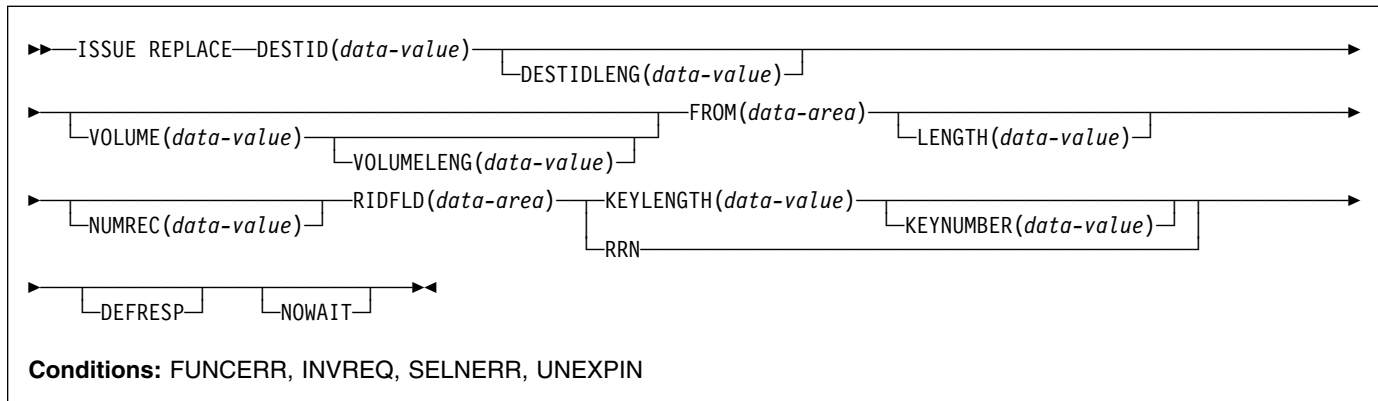
UNEXPIN

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

Relinquish use of a telecommunication line.

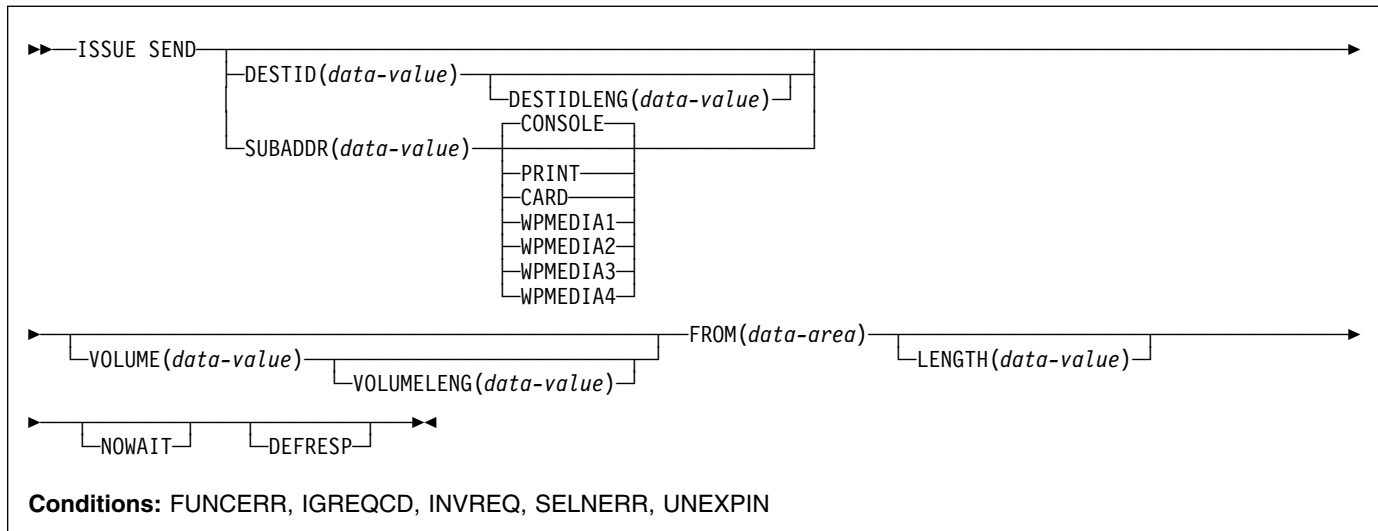
▶▶—ISSUE RESET—◀◀

Condition: INVREQ, NOTALLOC

This command is retained for compatability 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.

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.

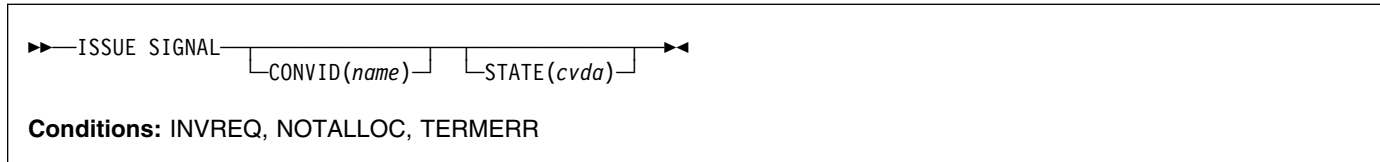
UNEXPIN

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.



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:

- The command has been used on an APPC conversation that is not using the EXEC CICS interface, or is not a mapped conversation.

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)

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.

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

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.

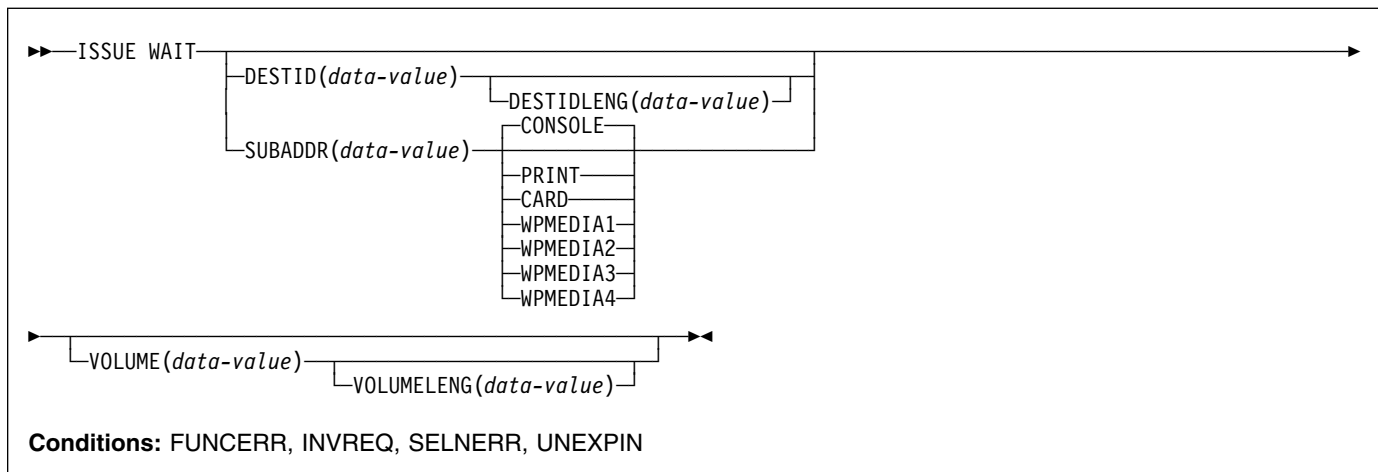
TERMERR

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.

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.

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

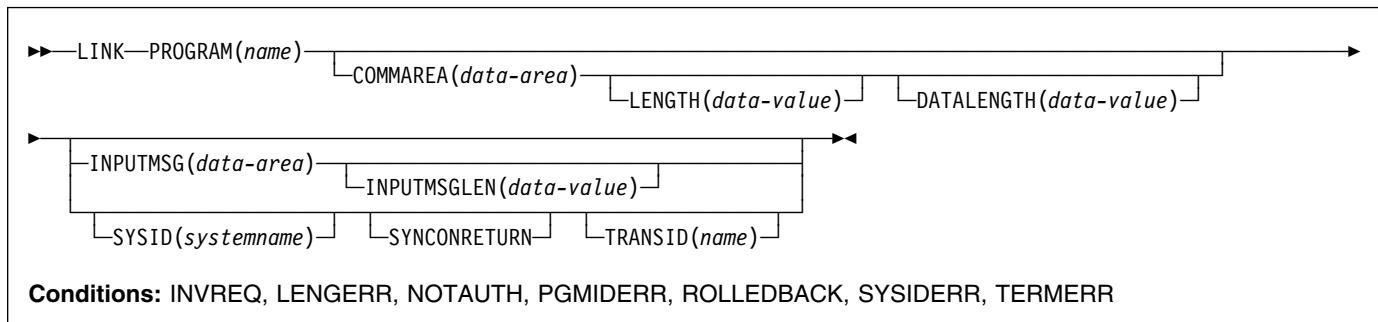
Create a journal record.

Note: This command is supported for compatibility with earlier releases of CICS. It is superseded by the WRITE JOURNALNAME command, which you are recommended to use instead.

LINK

Link to another program expecting return.

The external CICS interface (EXCI) provides a LINK command that performs all six commands of the interface in one invocation. See the *CICS External Interfaces Guide* for information about the EXCI.



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 this is a local definition, and the linked-to program is not already in main storage, CICS loads it.

In some circumstances, the linked-to program may reside on another CICS region—see “Distributed program link.”

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.

Distributed program link: In any of the following cases, the link is a *distributed program link* (DPL):

- You specify a remote region name on the SYSID option (with or without the associated TRANSID and SYNCONRETURN options).
- The REMOTESYSTEM option on the installed PROGRAM definition² specifies the name of a remote region.
- The installed program definition specifies DYNAMIC(YES)—or there is no installed program definition—and the dynamic routing program routes the link request to a remote region.

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 *CICS Messages and Codes* 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

² By “installed program definition” we mean a program definition that has been installed statically, by means of autoinstall, or by an EXEC CICS CREATE command.

- 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 403.

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.

Important

For examples of the use of the LINK command when the linked program is remote, see the *CICS Application Programming Guide*. For information about writing a dynamic routing program, see the *CICS Customization Guide*.

Options

COMMAREA(*data-area*)

specifies a communication area that is to be made available to the invoked program. In this option the data area is passed, and you must give it the name DFHCOMMAREA in the receiving program. (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). This value may not exceed 32 500 bytes if the COMMAREA is to be passed between any two CICS servers (for any combination of product/version/release).

PROGRAM(*name*)

specifies the identifier (1–8 characters) of the program to which control is to be passed unconditionally.

In any of the following cases, the linked-to program is a server program in a remote region:

- The SYSID option specifies a remote region.
- The REMOTESYSTEM option on the installed PROGRAM definition³ specifies the name of a remote region.
- The installed program definition specifies DYNAMIC(YES)—or there is no installed program definition—and the dynamic routing program routes the link request to a remote 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)
```

³ By “installed program definition” we mean a program definition that has been installed statically or dynamically by means of autoinstall, or by an EXEC CICS CREATE command.

LINK

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.

Changes to recoverable resources made by the server program are committed or rolled-back independently of changes to recoverable resources made by the client program issuing the **LINK** request, or changes made by the server in any subsequent **LINK**.

- The **NORMAL** condition is returned if changes to recoverable resources are committed before return from the server program.
- The **ROLLEDBACK** condition is returned if changes to recoverable resources are rolled back before return from the server program.
- The **TERMERR** condition is raised following failure of the communications link or the system in which the server program is executing. The client program is responsible for handling the condition and ensuring that data consistency is restored.

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.

If **SYSID** specifies a remote system, no reference is made to **PROGRAM** resource definitions held locally. If **SYSID** specifies the local system, CICS treats the **LINK** request as if **SYSID** was not specified.

A remote system name specified on the **SYSID** option takes priority over any remote system name returned by the dynamic routing program.

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**.

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 in the remote region specified on the **SYSID** option. (That is, a unit of work (**UOW**) is in progress, or the system initialization parameter **MROFSE=YES** has been specified in the client region.) 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.
- 17 The **TRANSID** supplied by the dynamic routing program 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**.
- 41 A **LINK** has been attempted to a Java program on the link stack.
- 43 A **LINK** has been attempted to a hot-pooled Java program object while there is already a hot-pooled program on the link stack.

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.

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

- 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 DFHRPL concatenation, 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.
- 25** The dynamic routing program rejected the link request.

Default action: terminate the task abnormally.

Note: RESP2 values are not returned to the client for conditions occurring in a DPL server program.

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.
- 20** The remote system specified by SYSID is an LUTYPE6.1-connected system. Distributed program link requests are not supported on LUTYPE6.1 connections.

Notes:

1. There is no local queuing in the event of a SYSIDERR.
 2. RESP2 values are not returned for conditions occurring on DPL requests.
- 28** The remote system specified by SYSID is not in service. This response can also indicate that the transaction has not been defined on the remote system.
- 29** The remote system specified by SYSID is in service, but there are no sessions available, and the dynamic routing program has chosen not to queue the link request.
- 31** The request to allocate a session to the remote system has been rejected.
- 32** The queue of allocate requests for sessions to the remote system has been purged.

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:

LINK

```
EXEC CICS LINK PROGRAM(PROGNAME)  
              COMMAREA(COMA) LENGTH(LENA)  
              DATALENGTH(LENI) SYSID('CONX')
```

LOAD

Load a program from the CICS DFHRPL concatenation library into main storage.

```

▶—LOAD—PROGRAM(name)—┐
                        └─SET(ptr-ref)—┐
                                ┌─LENGTH(data-area)—┐
                                └─FLENGTH(data-area)—┐
                                        ┌─ENTRY(ptr-ref)—┐
                                        └─HOLD—┐
                                                ◀◀
  
```

Conditions: INVREQ, LENGERR, NOTAUTH, PGMIDERR

Dynamic transaction routing

Using LOAD with HOLD, or using a resource that has been defined with RELOAD=YES, 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, table, or map. If the program is defined with RELOAD=NO, 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.

LOAD

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 DFHRPL concatenation, 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.
- 42** An attempt has been made to LOAD a JVM program. This is invalid because Java byte codes programs are not managed by CICS Loader.

Default action: terminate the task abnormally.

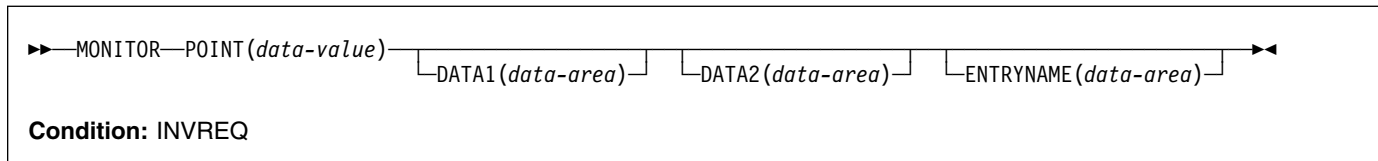
Examples

The following example shows how to load a user-prepared table called TB1:

```
EXEC CICS LOAD PROGRAM('TB1') SET(PTR)
```

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.
- 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 4 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)

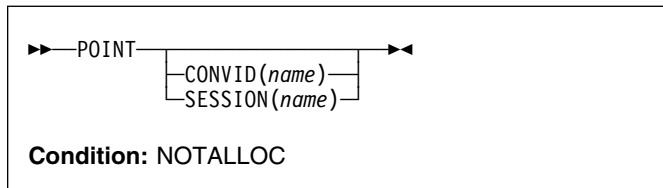
3:
EXEC CICS MONITOR
      POINT(13)
      DATA1(address of data)
      DATA2(length of data)
      ENTRYNAME(ENTRY3)
      needing: DFHMCT TYPE=EMP,
                CLASS=PERFORM,
                ID=(ENTRY3.13),
                PERFORM=MOVE(0,32)

```

Figure 4. 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

Restore the stack.



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

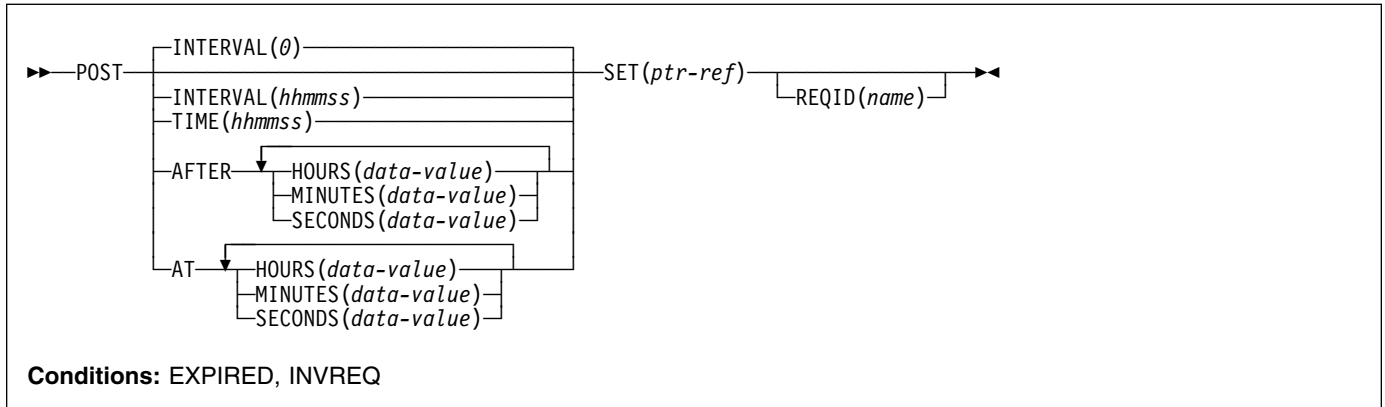
INVREQ

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).

Options

AFTER

specifies the interval of time to elapse.

There are two ways to enter the time under AFTER and AT.

POST

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

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 417 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

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.

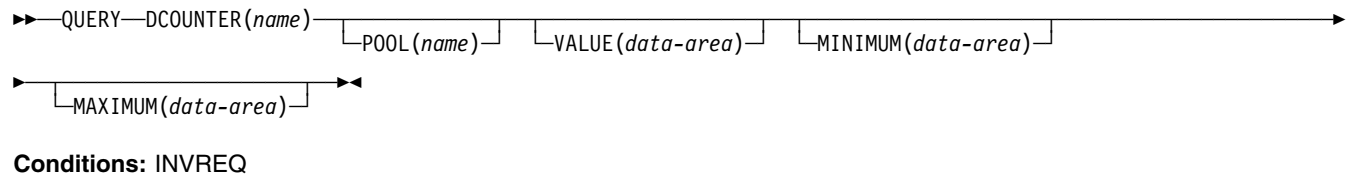
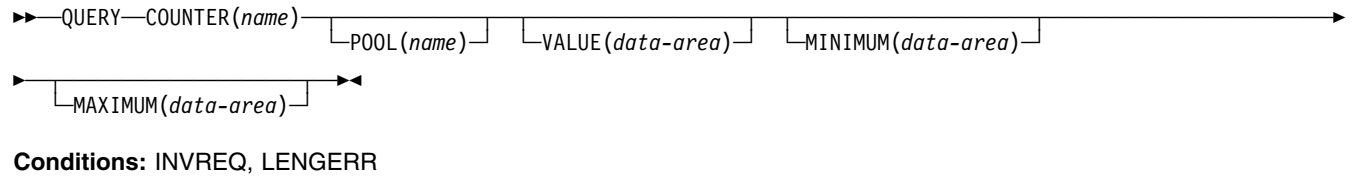
Note: When a CICS program uses EXEC CICS LINK to call another CICS program, the HANDLE CONDITION options are NOT inherited by the linked-to program, but CICS will search preceding logical levels for a HANDLE ABEND exit. See *CICS Application Programming Guide* for further details about the relationship 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 COUNTER

Query a named counter.



Description

These counter commands return the current, maximum, and minimum values for the named counter. COUNTER operates on fullword named counters and DCOUNTER operates on doubleword named counters.

For information about specifying fullword and doubleword variables on these named counter commands, see “Argument values” on page 2.

Options

COUNTER(*name*)

specifies the 16-character name of the fullword counter being queried. Valid characters for names are A through Z, 0 through 9, \$ @ # and _ (underscore). If *name* is a variable that contains a name that is less than 16 characters, the name must be padded with trailing blanks.

DCOUNTER(*name*)

specifies the 16-character name of the doubleword counter being queried. Valid characters for names are A through Z, 0 through 9, \$ @ # and _ (underscore). If *name* is a variable that contains a name that is less than 16 characters, the name must be padded with trailing blanks.

MAXIMUM(*data-area*)

specifies the data area in which CICS is to return the maximum number for the named counter. CICS returns a fullword signed binary value for the COUNTER command and a doubleword unsigned binary value for the DCOUNTER command

MINIMUM(*data-area*)

specifies the data area in which CICS is to return the minimum number for the named counter. CICS returns a fullword signed binary value for the COUNTER command and a doubleword unsigned binary value for the DCOUNTER command

POOL(*poolname*)

specifies the name of the pool in which the named counter resides.

Valid characters for the pool selector string are A through Z, 0 through 9, \$ @ # and _ (underscore). If *name* is a variable that contains a name that is less than 8 characters, the name must be padded with trailing blanks.

This parameter is optional. If you omit the name of the pool, a pool selector value of 8 blanks is assumed.

If there is no matching entry in the DFHNCOPT options table, CICS uses the name specified on the NCPLDFT system initialization parameter, which specifies the default named counter pool.

For information about generating a named counter options table using the DFHNCO macro, see the *CICS System Definition Guide*.

VALUE(*data-area*)

specifies the data area in which CICS is to return the current value for the named counter. CICS returns a fullword signed binary value for the COUNTER command and a doubleword unsigned binary value for the DCOUNTER command

QUERY COUNTER

Conditions

INVREQ

RESP2 values:

- 201** Named counter not found.
- 301** The server has reported an error code that is not understood by the named counter interface. Generally, this is not possible unless the interface load module, DFHNCIF, is at a lower maintenance or release level than the server itself.
- 303** An unexpected error, such as structure failure or loss of connectivity, has occurred on a macro used to access the coupling facility. Further information can be found in message DFHNC0441 in the CICS job log.
- 304** The pool selection parameter specified in the program cannot be resolved to a valid server name using the current options table.
- 305** The interface is unable to establish a connection to the server for the selected named counter pool. Further information can be found in an AXM services message (AXMSCnnnn) in the CICS job log.
- 306** An abend occurred during server processing of a request. Further information can be found in a message in the CICS job log and the server job log.
- 308** The DFHNCOPT options table module, required for resolving a pool name, could not be loaded.
- 309** During processing of the options table, the named counter interface encountered an unknown entry format. Either the options table is not correctly generated, or the DFHNCIF interface load module is not at the same release level as the options table.
- 310** An options table entry matching the given pool name specified a user exit program, but the user exit program is not link-edited with the options table and cannot be loaded.
- 403** The POOL parameter contains invalid characters or embedded spaces.
- 404** The COUNTER parameter contains invalid characters or embedded spaces.

Default action: terminate the task abnormally.

LENGERR

LENGERR occurs for COUNTER commands only and does not apply to DCOUNTER requests. It occurs when a counter that was defined by a DCOUNTER command has become negative when accessed as a fullword signed binary value (that is, the counter uses more than 31 bits).

In each of the three cases of overflow, the named counter server completes the operation, and returns a

warning response to CICS, which CICS returns to your application program as the RESP2 value. The data area contains the low-order 32 bits returned from the named counter server, which could be a negative number.

RESP2 values:

- 001** The current value that the server has attempted to return in one of the data areas has overflowed into the high-order (sign) bit (that is, the value returned is negative).

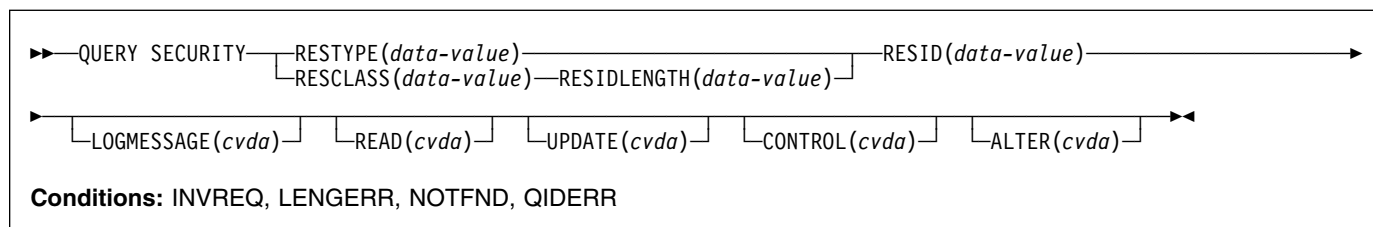
Note: LENGERR with RESP2=001 cannot occur for a named counter that is in the counter-at-limit condition. If the counter-at-limit condition has been reached, the value (which could be negative) is returned with a normal response.

- 002** A value is too large for a fullword data area by only 1 bit. In this case, the overflow value is exactly 1.
- 003** A value is too large for a fullword data area by a value greater than 1.

Default action: terminate the task abnormally.

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 RACF 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.

If the ESM is RACF, the class can be CICS-supplied or user-defined. 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 CICS or user-defined resource that you want to query the users access to. The value is a character string (1-12 characters for a CICS resource, and 1-246 characters for a user-defined resource, unless you are using the COBOL3 or OOCOBOL translator option in which case 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 RACF 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.

QUERY SECURITY

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:

```
DB2ENTRY
FILE
JOURNALNAME
JOURNALNUM2
PROGRAM
PSB
SPCOMMAND1
TDQUEUE
TRANSACTION
TRANSATTACH
TSQUEUE
```

¹ See *CICS RACF Security Guide*.

² Supported for compatibility with earlier releases.

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

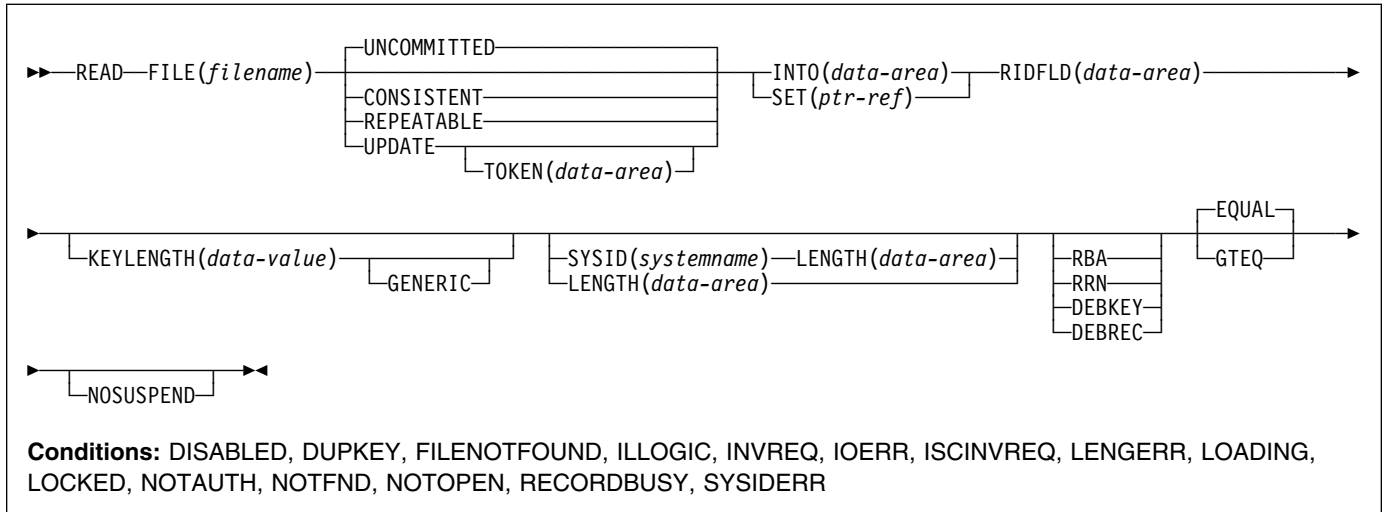
RESP2 values:

- 1 An indirect queue name associated with the given RESID is not found.

Default action: terminate the task abnormally.

READ

Read 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 (either a full key, RBA, or RRN). Immediately upon completion of a READ UPDATE command, the RIDFLD data area is available for reuse by the application program.

- # **Note:** You cannot specify an RBA for data sets greater than
- # 4GB (that is, defined with the extended format /
- # extended addressability attribute). The only data sets
- # greater than 4GB supported by CICS are KSDS, and
- # then only if they are accessed by key. CICS does not
- # support ESDS or RRDS data sets defined with the
- # extended attribute.

Data table considerations: 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 changed in any way.

| When this command reads a coupling-facility data table, only
| the data table is accessed, even if the table is initially loaded
| from a VSAM source data set.

- | **Note:** If a file that refers to a user-maintained or coupling
| facility data table is defined with RLSACCESS(YES),
| the RLS-specific API options CONSISTENT,
| NOSUSPEND, and REPEATABLE are not supported.

Reading files accessed in RLS mode: When a file is accessed in RLS mode, non-update read requests can specify one of the read integrity options: UNCOMMITTED, CONSISTENT, or REPEATABLE.

If none of these keywords is specified, CICS uses the value specified on the READINTEG parameter of the FILE resource definition, for which the default is UNCOMMITTED.

If you want to use the level of read integrity specified in the READINTEG keyword of the FILE definition, and then you need to change from using a local file to a remote file, or if you change the location of a remote file, ensure that:

- The remote file-owning region supports the read integrity options.
- The FILE definition in the remote system specifies:
 - RLS mode
 - The correct read integrity values for your application.

READ requests that specify the UPDATE keyword, or a CONSISTENT or REPEATABLE read integrity option, (either explicitly or implicitly in the FILE definition) return the LOCKED condition if they reference a record that has a retained lock. The key of a locked record is not returned to the application program. Thus, if an application program

READ

specifies GTEQ or GENERIC on the READ request it cannot tell which record key is locked.

If a request specifying read integrity is function-shipped to a member of the CICS family of products that does not support read integrity, the request fails:

- If an ISC link is used, the request receives an ATNI abend.
- If an MRO link is used, the request receives an AXF8 abend.

The AXF8 abend code indicates that your program has attempted to function-ship a request specifying file control options to a remote CICS region that does not support these options.

Options

CONSISTENT (RLS only)

The record is read with a level of read integrity provided by a VSAM shared lock that lasts for the duration of the request.

If the record is being modified by another task, which therefore holds an exclusive lock, the READ request waits until the update is complete (unless NOSUSPEND is also specified) as follows:

- For a READ request against a non-recoverable file, the READ completes as soon as any VSAM request performing the update completes.
- For a READ request against a recoverable file, the READ request completes when the updating task completes its next syncpoint or rollback.

DEBKEY

(blocked BDAM) specifies that deblocking is to occur by key. If neither DEBREC nor DEBKEY is specified, deblocking does not occur.

DEBREC

(blocked BDAM) 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 OS/VS COBOL and 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 READNEXT command, the LENGTH parameter contains the actual length of the record.

| This option must be specified if SYSID is specified.

| If the file is on a remote system, the LENGTH parameter need not be set here but must be set in the file resource definition

| If the file is on a local system, the LENGTH parameter must be set for variable-length records, using the INTO option, but not for fixed-length records. It is, however, advisable to specify the length for fixed-length records 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 for VS COBOL II, C, PL/I, and assembler-language applications 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.

| If you specify the SET option, you need not also specify
| the LENGTH option.

| 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 before truncation.

| 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.

NOSUSPEND (RLS only)

The request does not wait if the record is locked by VSAM with an active lock, including records locked as the result of a DEADLOCK.

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
- | • Coupling facility data tables
- # • KSDS and ESDS files that hold more than 4GB
- Any KSDS file opened in RLS access mode.

REPEATABLE (RLS only)

The record is read with a level of read integrity provided by a VSAM shared lock that lasts for the duration of the unit of work in which the read request is issued.

If the record is being modified by another task, which therefore holds an exclusive lock, the READ request waits until the update is complete (unless NOSUSPEND is also specified) as follows:

- For a recoverable file, the READ request completes when the updating transaction completes its next syncpoint or rollback.

- For a non-recoverable file, the READ completes as soon as the VSAM request performing the update completes.

After the READ request has completed, the record remains locked to the task that issued the READ. Other tasks may continue to read the record but no other task is allowed to update the record until the task that issued the READ performs its next syncpoint or rollback.

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 BDAM 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 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, the address of the data is below the 16MB line.

If TASKDATAKEY (USER) is specified for the executing transaction, the data returned is in a user-key; otherwise it is in a CICS-key.

READ

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.

TOKEN can be function shipped. However, if a request specifying TOKEN is function shipped to a member of the CICS family of products that does not recognize this keyword, the request fails.

Note: If you specify TOKEN it implies update.

UNCOMMITTED

The record is read without read integrity.

The current value of the record, as known to VSAM, is returned. No attempt is made to serialize this read request with any concurrent update activity for the same record. The record may be in the process of being updated by another task, and the record data may change later if that update is subsequently backed out.

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. The mechanism used to ensure data integrity depends on the type of file resource:

- For a VSAM file accessed in RLS mode, the record to be updated is locked by the SMSVSAM server.
- For a VSAM file accessed in non-RLS mode, the record to be updated is locked by CICS and, in addition, the control interval containing the record is held in exclusive control by VSAM.
- For a BDAM file, the record to be updated is held in exclusive control by BDAM.
- For a user-maintained data table, the record to be updated is locked by CICS.
- For a CICS-maintained data table, the record to be updated is locked by CICS and, in addition, the control interval containing the record is held in exclusive control by VSAM. The VSAM control interval lock is required because changes to the data table are reflected in the source data set, which is accessed in non-RLS mode.
- For a coupling facility data table using the locking model, the record to be updated is locked by the coupling facility data table server.

- For a coupling facility data table using the contention model, records are not locked, enabling the records to be read for update by more than one task. If a record read for update by one task is then changed by another, the first task is notified, when it issues a REWRITE or DELETE command, by the CHANGED exception condition. If a record read for update by one task is then deleted by another, the first task is notified, when it issues a REWRITE or DELETE command, by the NOTFND condition.

If another task has issued a READ REPEATABLE request against the same record, your READ UPDATE is made to wait until that task reaches SYNCPOINT (unless you issued NOSUSPEND).

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 377.)

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.
- 28 Following a READ UPDATE command without TOKEN, another READ UPDATE without TOKEN was issued against the same file without an intervening REWRITE, DELETE, UNLOCK or SYNCPOINT command. This condition may in some cases be raised despite the fact that the first READ UPDATE was not successful, for example because it timed out
- 40 A BDAM 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 or coupling facility data table; for example, if RBA is specified.
- 51 A READ to a KSDS file that is being accessed in RLS mode specifies the RBA keyword. RLS mode does not support relative byte address access to KSDS data sets.
- 52 CONSISTENT is specified on a READ request to a non-RLS mode file, or to a data table that is specified with RLSACCESS(YES). CONSISTENT is not allowed if the file refers to a data table, even if the file definition specifies RLSACCESS(YES).
- 53 REPEATABLE is specified on a READ request to a non-RLS mode file or to a data table that is specified with RLSACCESS(YES). REPEATABLE is not allowed if the file refers to a data table, even if the file definition specifies RLSACCESS(YES).
- 55 NOSUSPEND is specified on a READ request to a non-RLS mode file or to a data table that is specified with RLSACCESS(YES). NOSUSPEND is not allowed if the file refers to a data table, even if the file definition specifies RLSACCESS(YES).
- 56 An attempt to update a recoverable coupling facility data table has failed because the current unit of work has already updated 1024 recoverable

coupling facility data tables. You cannot update more than 1024 recoverable coupling facility data tables within a unit of work

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.
For a coupling facility data table, an IOERR indicates a bad response returned from a coupling facility access.
(Further information is available in the EXEC interface block, defined on 377.)

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 option nor the SET option is specified on a READ command for a file with variable-length records or for a BDAM 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 The request cannot be satisfied because it is issued against a data table that is still being loaded. The condition can be raised for one of the following reasons:
 - The READ specifies a record that has not yet been loaded into a coupling facility data table. Records can be read or modified while a CFDT is loading only if the requested key is

READ

within the range of those records already loaded.

The LOADING response can also be returned for a coupling facility data table that has failed during loading. For more information about what happens if the load for a coupling facility data table fails, see the description of the XDTLC global user exit in the *CICS Customization Guide*.

- The READ specifies the UPDATE option for a user-maintained data table. A user-maintained data table cannot be modified during loading.
- The READ specifies the GENERIC or GTEQ options for a user-maintained data table. While a UMT is being loaded, you can use read requests with precise keys only.

If your application programs encounter the LOADING condition persistently or too frequently, check that this is not caused by conflicting file definitions that reference the same data set.

Default action: terminate the task abnormally.

LOCKED

RESP2 values:

- 106** An attempt is being made to read a record either specifying the UPDATE keyword, or specifying (explicitly or implicitly) CONSISTENT or REPEATABLE, but the record is locked by a retained lock (see “Retained and active locks” on page 203).

The LOCKED condition can also occur for a READ UPDATE request to a recoverable CFDT that uses the locking model, if the record being read is locked by a retained lock. See the *CICS Recovery and Restart Guide* for information about investigating retained locks on records in a coupling facility data table.

Default action: abend the task with code AEX8.

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 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 (if the table was created from

one); 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 data table.

Default action: terminate the task abnormally.

NOTOPEN

RESP2 values:

- 60** NOTOPEN (RESP2 60) is returned for one of the following reasons:

- 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. You can also make CLOSED, UNENABLED the initial state, by specifying STATUS(UNENABLED) and OPENTIME(FIRSTREF) on the FILE resource definition. (For BDAM files, you use the FILSTAT parameter of the DFHFCT TYPE=FILE macro.)
- The requested file is OPEN and in use by other transactions, but a CLOSE request against the file has been received.
- A READ command is issued against a data set that is quiesced, or is being quiesced, as a result of a SET DSNNAME QUIESCED or IMMQUIESCED command.

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.

RECORDBUSY

RESP2 values:

- 107** The NOSUSPEND keyword is specified and the record is locked by an active lock (see “Retained and active locks” on page 203).

Default action: abend the task with code AEX9.

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.
- 131** For a coupling facility data table, the connection to the coupling facility data table server has failed. This could be because the server itself has failed, or the server is available, but CICS has failed to connect to it.

| **132** The READ is issued for a coupling facility data
 | table that no longer exists, probably because of a
 | coupling facility failure, in which case the coupling
 | facility data table server also fails. See the *CICS*
 | *System Definition Guide* for information about
 | restarting a coupling facility data table server and
 | reloading a table.

Default action: terminate the task abnormally.

Retained and active locks: RECORDBUSY refers to active locks and LOCKED refers to retained locks:

These locks affect READ requests which acquire locks; that is, update requests and requests with read integrity. These are the kinds of READ requests which are referred to in the following bullets. Other READ requests are unaffected by the presence of retained or active locks.

- READ requests for records that have *retained* locks are always rejected with a LOCKED response.
- READ requests for records that have *active* locks wait for the lock to be released, except when the NOSUSPEND keyword is specified, in which case CICS returns the RECORDBUSY response.

Examples

The following example shows you how to read a record from a file named MASTER into a specified data area:

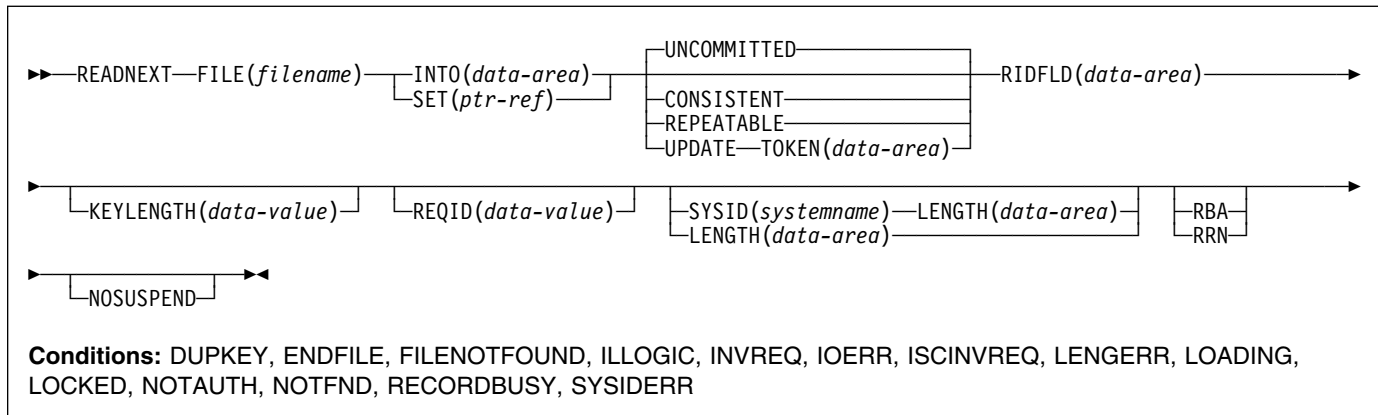
```
EXEC CICS READ
  INTO(RECORD)
  FILE('MASTER')
  RIDFLD(ACCTNO)
```

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('MASTVSAM')
  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.

- # **Note:** You cannot specify an RBA for data sets greater than
- # 4GB (that is, defined with the extended format /
- # extended addressability attribute). The only data sets
- # greater than 4GB supported by CICS are KSDS, and
- # then only if they are accessed by key. CICS does
- # not support ESDS or RRDS data sets defined with
- # the extended attribute.

RIDFLD 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 BDAM, 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).

Reading files accessed in RLS mode: For files accessed in RLS mode, you can include the UPDATE keyword on the READNEXT request to update some records during the browse. If you specify UPDATE you must also specify TOKEN. You can then update the record by issuing a DELETE or REWRITE command specifying the TOKEN returned on the browse function.

Note: TOKEN without the UPDATE keyword implies UPDATE.

Use of the UPDATE option is subject to the following rules:

- You can specify UPDATE on a READNEXT command only if the file is accessed in RLS mode. If you specify UPDATE for a file accessed in non-RLS mode, CICS returns INVREQ.
- You can specify UPDATE on READNEXT, but not on the STARTBR or RESETBR commands.
- You can intermix UPDATE and non-update requests within the same browse.
- CICS does not preserve the UPDATE option for you from one READNEXT command to the next.

CICS supports only one TOKEN in a browse sequence, and the TOKEN value on each READNEXT invalidates the previous value.

Locks for UPDATE: Specifying UPDATE on a READNEXT acquires an exclusive lock. The duration of these exclusive locks within a browse depends on the action your application program takes:

- If you decide to DELETE or REWRITE the last record acquired by a READNEXT UPDATE in a browse, using the associated token, the lock remains active as follows:
 - If the file is recoverable, the lock is released at completion of the next syncpoint or rollback.
 - If the file is non-recoverable, the lock will be released by the time ENDBR has completed, but might be released earlier.
- If you decide *not* to update the last record read, CICS frees the exclusive lock when your program either issues another READNEXT or READPREV command, or ends the browse.

UNLOCK note

The UNLOCK command does *not* free an exclusive lock held by VSAM against a record acquired by READNEXT UPDATE. An UNLOCK in a browse invalidates the TOKEN returned by the last request.

Locks for read integrity: Specifying one of the read integrity options acquires a shared lock on each READNEXT. The duration of these shared locks with a browse depends on the type of read integrity you specify:

- If you specify CONSISTENT read integrity, the shared lock is held only for the duration of each read request—that is, until the record is returned to your program.
- If you specify REPEATABLE read integrity, the shared lock is held for the duration of the unit of work in which the browse is performed. In this case, your program could acquire a large number of shared locks, which will prevent the granting of exclusive locks for update functions. You should use REPEATABLE read integrity in a browse with caution.

Function shipping READNEXT with UPDATE or read-integrity: If a READNEXT command specifying UPDATE or one of the read-integrity options is function-shipped to a member of the CICS family of products that does not support UPDATE or the read integrity options, the request fails:

- If an ISC link is used, the request receives an ATNI abend.
- If an MRO link is used, the request receives an AXF8 abend.

AXF8 is an abend code, received by the sending side of a function-shipped request. It indicates that an attempt has been made to send a request specifying UPDATE on an MRO link to a CICS region that does not support update or read integrity options.

Read integrity: When a file is accessed in RLS mode, non-update read requests can specify read integrity options: UNCOMMITTED, CONSISTENT, or REPEATABLE.

If you don't specify any of these keywords, CICS uses the value specified on the READINTEG parameter of the FILE resource definition, for which the default is UNCOMMITTED.

If you want to use the level of read integrity specified in the READINTEG keyword of the FILE definition, and then you need to change from using a local file to a remote file, or if you change the location of a remote file, ensure that:

- The remote file-owning region is at the CICS Transaction Server for OS/390 Release 1 (or later) level.
- The FILE definition in the remote system specifies:
 - RLS mode
 - The correct read integrity values for your application.

Options

CONSISTENT (RLS only)

The record is read with a level of read integrity provided by a VSAM shared lock that lasts for the duration of the request.

If the record is being modified by another task, which therefore holds an exclusive lock, the READNEXT request waits until the update is complete (unless NOSUSPEND is also specified) as follows:

- For a READNEXT request against a non-recoverable file, the READ completes as soon as any VSAM request performing the update completes.
- For a READNEXT request against a recoverable file, the READ request completes when the updating task completes its next syncpoint or rollback.

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

READNEXT

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. 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 data area where the record is to be put. On completion of the READNEXT command, the LENGTH parameter contains the actual length of the record.

| This option must be specified if SYSID is specified.

| If the file is on a remote system, the LENGTH parameter need not be set here but must be set in the file resource definition

| If the file is on a local system, the LENGTH parameter must be set for variable-length records, using the INTO option, but not for fixed-length records. It is, however, advisable to specify the length for fixed-length records 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 for VS COBOL II, C, PL/I, and assembler-language applications 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.

| If you specify the SET option, you need not also specify the LENGTH option.

| 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 before truncation.

| 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.

NOSUSPEND (RLS only)

The request does not wait if the record is locked by VSAM with an active lock, including records locked as the result of a DEADLOCK.

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
- Coupling facility data tables
- Any KSDS files opened in RLS access mode
- KSDS or ESDS files that hold more than 4GB.

REPEATABLE (RLS only)

The record is read with a level of read integrity provided by a VSAM shared lock that lasts for the duration of the unit of work in which the read request is issued.

If the record is being modified by another task, which therefore holds an exclusive lock, the READNEXT request waits until the update is complete (unless NOSUSPEND is also specified) as follows:

- For a recoverable file, the READNEXT request completes when the updating transaction completes its next syncpoint or rollback.
- For a non-recoverable file, the READNEXT completes as soon as the VSAM request performing the update completes.

After the READNEXT request has completed, the record remains locked to the task that issued the READNEXT. Other tasks may continue to read the record but no other task is allowed to update the record until the task that issued the READNEXT performs its next syncpoint or rollback.

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 BDAM 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 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, the address returned in the SET pointer is below the 16MB line.

If TASKDATAKEY(USER) is specified for the running task, the data returned is in user-key storage; otherwise it is in CICS-key storage.

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.

TOKEN(data-area) (RLS only)

Returns, as a fullword binary value, the request identifier for this READNEXT 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 READNEXT command.

You must specify the returned TOKEN on a subsequent REWRITE or DELETE command to identify the record being rewritten or deleted. You can also specify the value returned by CICS on the TOKEN option on a subsequent UNLOCK command, to identify the token that is being invalidated.

You must specify TOKEN if you specify UPDATE (but UPDATE is assumed if you specify TOKEN without UPDATE).

CICS supports only one active TOKEN at a time for a given REQID. Thus, a TOKEN value remains valid only until the following READNEXT, READPREV, or ENDBR command for this browse, or until a REWRITE, DELETE, or UNLOCK command (whichever is first).

TOKEN can be function shipped. However, if a request specifying TOKEN is function shipped to a member of the CICS family of products that does not recognize this keyword, the request fails.

UNCOMMITTED

The record is read without read integrity. The current data, as known to VSAM, is returned. No attempt is made to serialize this read request with any concurrent update activity for the same record. The record may be being updated by another transaction, therefore the value of the record may change later if that update is subsequently backed out.

UPDATE (RLS only)

Specifies that the record is to be obtained for updating or deletion. If this option and TOKEN are both omitted, read only is assumed.

READNEXT

If you specify UPDATE, you must also specify TOKEN.

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 377.)

Default action: terminate the task abnormally.

INVREQ

RESP2 values:

- 20** The FILE definition does not allow updates.
- 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.
- 52** CONSISTENT is specified on a READ request to a non-RLS mode file, or to a data table that is specified with RLSACCESS(YES). CONSISTENT is not allowed if the file refers to a data table, even if the file definition specifies RLSACCESS(YES).
- 53** REPEATABLE is specified on a READ request to a non-RLS mode file, or to a data table that is specified with RLSACCESS(YES). REPEATABLE is not allowed if the file refers to a data table, even if the file definition specifies RLSACCESS(YES).
- 54** UPDATE is not allowed because the file is not a VSAM file that is accessed in RLS mode.
- 55** NOSUSPEND is specified on a READ request to a non-RLS mode file, or to a data table that is specified with RLSACCESS(YES). NOSUSPEND is not allowed if the file refers to a data table, even if the file definition specifies RLSACCESS(YES).

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.

For a coupling facility data table, an IOERR indicates a bad response returned from a coupling facility access.

Further information is available in the EXEC interface block, defined on 377.

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 BDAM 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.

LOADING

RESP2 values:

- 104 The read request specifies a record key for a record in a coupling facility data table that is still being loaded, and the key is out of range of the records already loaded. Records can be browsed in a coupling facility data table during loading only if the requested key is within the range of those records already loaded.

The LOADING response can also be returned for a coupling facility data table that has failed during loading. For more information about what happens if the load for a coupling facility data table fails, see the description of the XDTLC global user exit in the *CICS Customization Guide*.

If your application programs encounter the LOADING condition persistently or too frequently, check that this is not caused by conflicting file definitions that reference the same data set.

LOCKED

RESP2 values:

- 106 The read request specifies the UPDATE keyword, or one of the read integrity keywords CONSISTENT or REPEATABLE, or the file resource definition specifies read integrity, but VSAM holds a retained lock against the record (see “Retained and active locks” on page 210).

The key of the locked record is not returned to your application program. If you handle this condition and control is returned to your program, the browse can continue and retrieve the record following the locked record by issuing another READNEXT request.

The LOCKED condition can also occur for a request to a recoverable CFDT that uses the locking model, if the record being read is locked by a retained lock. See the *CICS Recovery and Restart Guide* for information about investigating retained locks on records in a coupling facility data table.

Default action: abend the task with code AEX8.

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.

RECORDBUSY

RESP2 values:

- 107 NOSUSPEND is specified on the request but VSAM holds an active lock against the record, which would cause the request to wait (see “Retained and active locks” on page 210).

The key of the locked record is not returned to your application program. If you handle this condition and control is returned to your program, the browse can continue and retrieve the record following the locked record by issuing another READNEXT request.

Default action: abend the task with code AEX9.

SYSIDERR

RESP2 values:

- 130 The SYSID option specifies a name that is neither the local CICS region nor a remote system (made known to CICS by defining a CONNECTION). SYSIDERR also occurs when the link to the remote system is closed
- 131 For a coupling facility data table, the connection to the coupling facility data table server has failed. This could be because the server itself has failed, or the server is available, but CICS has failed to connect to it.
- 132 The READNEXT is issued against a coupling facility data table that no longer exists, probably because of a coupling facility failure, in which case the coupling facility data table server also fails. See the *CICS System Definition Guide* for information about restarting a coupling facility data table server and reloading a table.

Default action: terminate the task abnormally.

READNEXT

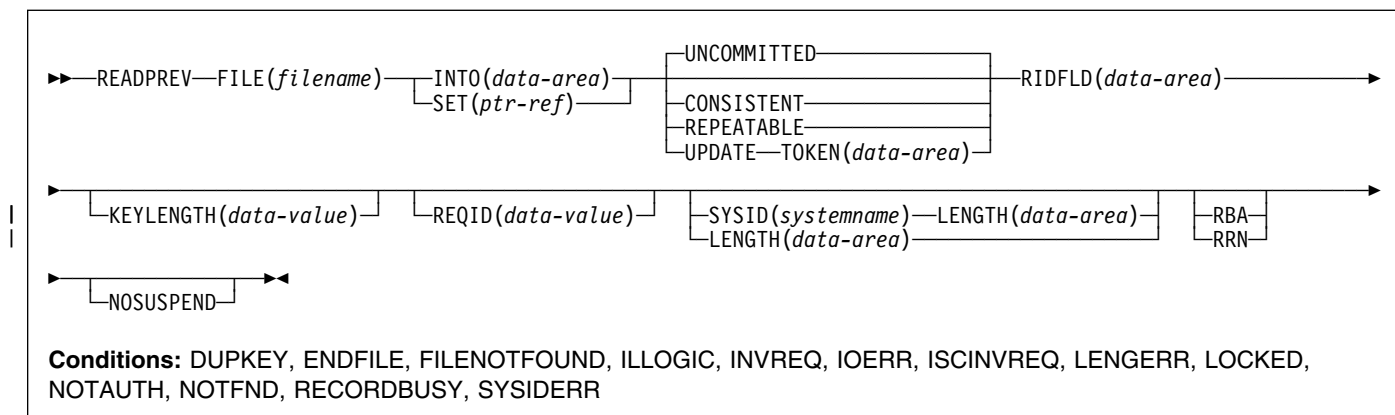
Retained and active locks: RECORDBUSY refers to active locks, and LOCKED refers to retained locks:

These locks affect READNEXT requests which acquire locks; that is, update requests and requests with read integrity. These are the kinds of READNEXT requests which are referred to in the following bullets. Other READNEXT requests are unaffected by the presence of retained or active locks.

- READNEXT requests for records that have *retained* locks are always rejected with a LOCKED response.
- READNEXT requests for records that have *active* locks wait for the lock to be released, except when the NOSUSPEND keyword is specified, in which case CICS returns the RECORDBUSY response.

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.

Note: You cannot specify an RBA for data sets greater than
 # 4GB (that is, defined with the extended format /
 # extended addressability attribute). The only data sets
 # greater than 4GB supported by CICS are KSDS, and
 # then only if they are accessed by key. CICS does not
 # support ESDS or RRDS data sets defined with the
 # extended attribute.

RIDFLD 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).

Reading files accessed in RLS mode: For files accessed in RLS mode, you can include the UPDATE keyword on the READPREV request to update some records during the browse. If you specify UPDATE you must also specify TOKEN. You can then update the record by issuing a DELETE or REWRITE command specifying the TOKEN returned on the browse function.

Note: TOKEN without the UPDATE keyword implies UPDATE.

Use of the UPDATE option is subject to the following rules:

- You can specify UPDATE on a READPREV command only if the file is accessed in RLS mode. If you specify UPDATE for a file accessed in non-RLS mode, CICS returns INVREQ.
- You can specify UPDATE on READPREV, but not on the STARTBR or RESETBR commands.
- You can intermix UPDATE and non-update requests within the same browse.
- CICS does not preserve the UPDATE option for you from one READPREV command to the next.

READPREV

CICS supports only one TOKEN in a browse sequence, and the TOKEN value on each READPREV invalidates the previous value.

Locks for UPDATE: Specifying UPDATE on a READPREV acquires an exclusive lock. The duration of these exclusive locks within a browse depends on the action your application program takes:

- If you decide to DELETE or REWRITE the last record acquired by a READPREV UPDATE in a browse, using the associated token, the lock remains active as follows:
 - If the file is recoverable, the lock is released at completion of the next syncpoint or rollback.
 - If the file is non-recoverable, the lock will be released by the time ENDBR has completed, but might be released earlier.
- If you decide *not* to update the last record read, CICS frees the exclusive lock when your program either issues another READNEXT or READPREV command, or ends the browse.

UNLOCK note

The UNLOCK command does *not* free an exclusive lock held by VSAM against a record acquired by READPREV UPDATE. An UNLOCK in a browse invalidates the TOKEN returned by the last request.

Locks for read integrity: Specifying one of the read integrity options acquires a shared lock on each READPREV. The duration of these shared locks with a browse depends on the type of read integrity you specify:

- If you specify CONSISTENT read integrity, the shared lock is held only for the duration of each read request—that is, until the record is returned to your program.
- If you specify REPEATABLE read integrity, the shared lock is held for the duration of the unit of work in which the browse is performed. In this case, your program could acquire a large number of shared locks, which will prevent the granting of exclusive locks for update functions. You should use REPEATABLE read integrity in a browse with caution.

Function shipping READPREV with UPDATE or read-integrity: If a READPREV command specifying UPDATE or one of the read-integrity options is function-shipped to a member of the CICS family of products that does not support UPDATE or the read integrity options, the request fails:

- If an ISC link is used, the request receives an ATNI abend.
- If an MRO link is used, the request receives an AXF8 abend.

AXF8 is an abend code, received by the sending side of a function-shipped request. It indicates that an attempt

has been made to send a request specifying UPDATE on an MRO link to a CICS region that does not support update or read integrity options.

Read integrity: When a file is accessed in RLS mode, non-update read requests can specify read integrity options: UNCOMMITTED, CONSISTENT, or REPEATABLE.

If you don't specify any of these keywords, CICS uses the value specified on the READINTEG parameter of the FILE resource definition, for which the default is UNCOMMITTED.

If you want to use the level of read integrity specified in the READINTEG keyword of the FILE definition, and then you need to change from using a local file to a remote file, or if you change the location of a remote file, ensure that:

- The remote file-owning region is at the CICS Transaction Server for OS/390 Release 1 (or later) level.
- The FILE definition in the remote system specifies:
 - RLS mode
 - The correct read integrity values for your application.

Options

CONSISTENT (RLS only)

The record is read with a level of read integrity provided by a VSAM shared lock that lasts for the duration of the request.

If the record is being modified by another task, which therefore holds an exclusive lock, the READPREV request waits until the update is complete (unless NOSUSPEND is also specified) as follows:

- For a READPREV request against a non-recoverable file, the READPREV completes as soon as any VSAM request performing the update completes.
- For a READPREV request against a recoverable file, the READPREV request completes when the updating task completes its next syncpoint or rollback.

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 data area where the record is to be put. On completion of the READNEXT command, the LENGTH parameter contains the actual length of the record.

This option must be specified if SYSID is specified.

If the file is on a remote system, the LENGTH parameter need not be set here but must be set in the file resource definition

If the file is on a local system, the LENGTH parameter must be set for variable-length records, using the INTO option, but not for fixed-length records. It is, however, advisable to specify the length for fixed-length records 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 for VS COBOL II, C, PL/I, and assembler-language applications 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.

If you specify the SET option, you need not also specify the LENGTH option.

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 before truncation.

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.

NOSUSPEND (RLS only)

The request does not wait if the record is locked by VSAM with an active lock, including records locked as the result of a DEADLOCK.

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 STARTBTR or RESETBTR command that specified the RBA option.

You cannot use RBA for:

- User-maintained data tables
- Coupling facility data tables
- Any KSDS files opened in RLS access mode
- KSDS or ESDS files that hold more than 4GB

REPEATABLE (RLS only)

The record is read with a level of read integrity provided by a VSAM shared lock that lasts for the duration of the unit of work in which the read request is issued.

If the record is being modified by another task, which therefore holds an exclusive lock, the READPREV request waits until the update is complete (unless NOSUSPEND is also specified) as follows:

- For a recoverable file, the READPREV request completes when the updating transaction completes its next syncpoint or rollback.
- For a non-recoverable file, the READPREV completes as soon as the VSAM request performing the update completes.

After the READPREV request has completed, the record remains locked to the task that issued the READPREV. Other tasks can continue to read the record, but no other task is allowed to update the record until the task that issued the READPREV performs its next syncpoint or rollback.

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.

READPREV

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 associated with the application program, the address returned in the SET pointer can be above or below the 16MB line.

If DATALOCATION(BELOW) is associated with the application program, the address returned in the SET pointer is below the 16MB line.

If TASKDATAKEY(USER) is specified for the running task, the data returned is in user-key storage; otherwise it is in CICS-key storage.

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.

TOKEN(*data-area*) (RLS only)

Returns, as a fullword binary value, a unique identifier for this READPREV 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 READPREV command.

Your application program must specify the returned TOKEN on a subsequent REWRITE or DELETE command to identify the record being rewritten or deleted. Your application program can also specify the returned TOKEN on a subsequent UNLOCK command to identify the token that is being invalidated.

You must specify TOKEN if you specify UPDATE (but UPDATE is assumed if you specify TOKEN without UPDATE).

CICS supports only one active TOKEN at a time for a given REQID. Thus, a TOKEN value remains valid only until the following READNEXT, READPREV, or ENDBR command for this browse, or until a REWRITE, DELETE, or UNLOCK command (whichever is first).

TOKEN can be function shipped. However, if a request specifying TOKEN is function shipped to a member of the CICS family of products that does not recognize this keyword, the request fails.

UNCOMMITTED

The record is read without read integrity. The current data, as known to VSAM, is returned. No attempt is made to serialize this read request with any concurrent update activity for the same record. The record may be being updated by another transaction, therefore the

value of the record may change later if that update is subsequently backed out.

UPDATE (RLS only)

Specifies that the record is to be obtained for updating or deletion. If this option and TOKEN are both omitted, read only is assumed.

If you specify UPDATE, you must also specify TOKEN.

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 377.)

Default action: terminate the task abnormally.

INVREQ

RESP2 values:

20 The FILE definition does not allow updates.

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 BDAM file.
- 41 The REQID, if any, does not match that of any successful STARTBR command.
- 52 CONSISTENT is not allowed because the file is not a VSAM file that is accessed in RLS mode.
- 53 REPEATABLE is not allowed because the file is not a VSAM file that is accessed in RLS mode.
- 54 UPDATE is not allowed because the file is not a VSAM file that is accessed in RLS mode.
- 55 NOSUSPEND is not allowed because the file is not a VSAM file that is accessed in RLS mode.

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 377.)

| For a coupling facility data table, an IOERR
| indicates a bad response returned from a coupling
| facility access.

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.

LOCKED

RESP2 values:

- 106 The read request specifies the UPDATE keyword, or one of the read integrity keywords CONSISTENT or REPEATABLE, or the file resource definition specifies read integrity, but VSAM holds a retained lock against the record (see "Retained and active locks" on page 216).

The key of the locked record is not returned to your application program. If you handle this condition and control is returned to your program, the browse can continue and retrieve the record following the locked record by issuing another READPREV request.

| The LOCKED condition can also occur for a
| request to a recoverable CFDT that uses the
| locking model, if the record being read is locked
| by a retained lock. See the *CICS Recovery and
| Restart Guide* for information about investigating
| retained locks on records in a coupling facility data
| table.

Default action: abend the task with code AEX8.

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.

RECORDBUSY

RESP2 values:

- 107 NOSUSPEND is specified on the request but VSAM holds an active lock against the record, which would cause the request to wait (see "Retained and active locks" on page 216).

The key of the locked record is not returned to your application program. If you handle this condition and control is returned to your program, the browse can continue and retrieve the record following the locked record by issuing another READPREV request.

Default action: abend the task with code AEX9.

SYSIDERR

RESP2 values:

READPREV

- 130** The SYSID option specifies a name that has not been defined to CICS as a remote system (defined by defining a CONNECTION). SYSIDERR also occurs when the link to the remote system is closed.
- | **131** For a coupling facility data table, the connection to
| the coupling facility data table server has failed.
| This could be because the server itself has failed,
| or the server is available, but CICS has failed to
| connect to it.
- | **132** The READPREV is issued against a coupling
| facility data table that no longer exists, probably
| because of a coupling facility failure, in which case
| the coupling facility data table server also fails.
| See the *CICS System Definition Guide* for
| information about restarting a coupling facility data
| table server and reloading a table.

Default action: terminate the task abnormally.

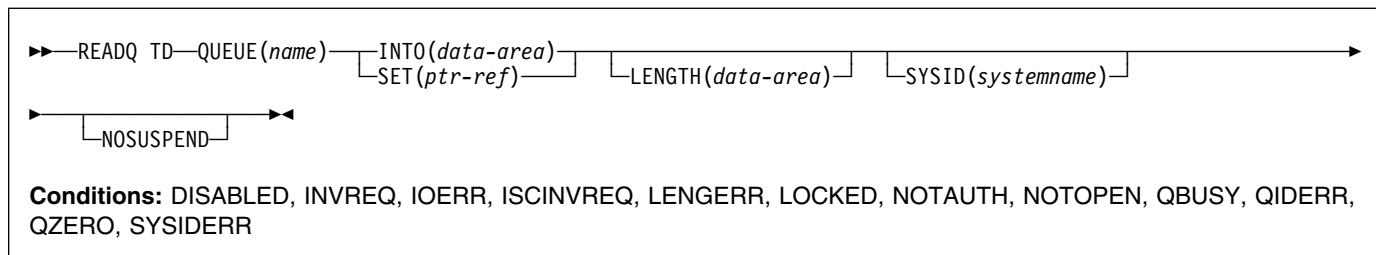
Retained and active locks: RECORDBUSY refers to active locks and LOCKED refers to retained locks:

These locks affect READPREV requests which acquire locks; that is, update requests and requests with read integrity. These are the kinds of READPREV requests which are referred to in the following bullets. Other READPREV requests are unaffected by the presence of retained or active locks.

- READPREV requests for records that have *retained* locks are always rejected with a LOCKED response.
- READPREV requests for records that have *active* locks wait for the lock to be released, except when the NOSUSPEND keyword is specified, in which case CICS returns the RECORDBUSY response.

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 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

READQ TD

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

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.

LOCKED

occurs when the request cannot be performed because use of the queue has been restricted owing to a unit of work failing in-doubt. This can happen on any request for a logically-recoverable queue defined with WAIT(YES) and WAITACTION(REJECT) in the TDQUEUE resource definition.

Specify WAIT(YES) and WAITACTION(Queue) in the TDQUEUE resource definition if you want the transaction to wait.

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:

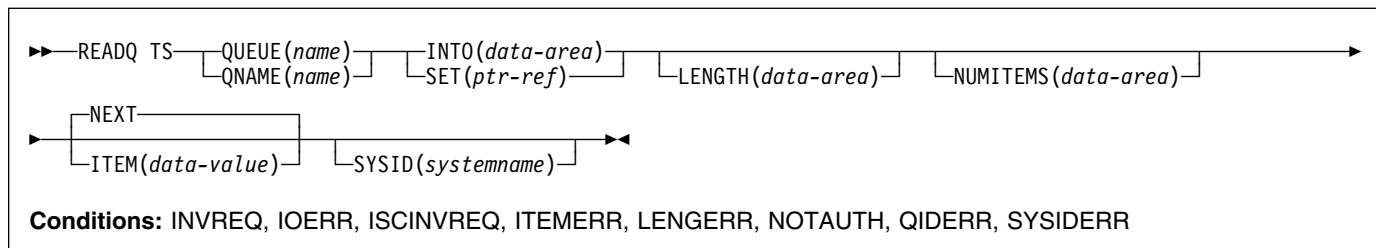
```
EXEC CICS READQ TD
      QUEUE('CSML')
      INTO(DATA)
      LENGTH(LDATA)
```

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.

QNAME(*name*)

an alternative to QUEUE, QNAME specifies the symbolic name (1–16 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 the application program is defined with DATALLOCATION(ANY), the address of the data can be above or below the 16MB line. If the application program is defined with DATALLOCATION(BELOW), the address of the data is below the 16MB line.

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.

READQ TS

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

RESP2 values:

5 There is an irrecoverable input/output error for a shared queue.

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

RESP2 values:

101 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

RESP2 values:

4 The CICS region in which the temporary storage command is executed fails to connect to the TS server managing the TS pool that supports the referenced temporary storage queue. (For example, this can happen if the CICS region is not authorized to access the temporary storage server).

SYSIDERR can also occur if the temporary storage server has not been started, or because the server has failed (or been stopped) while CICS continues executing. Also 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.

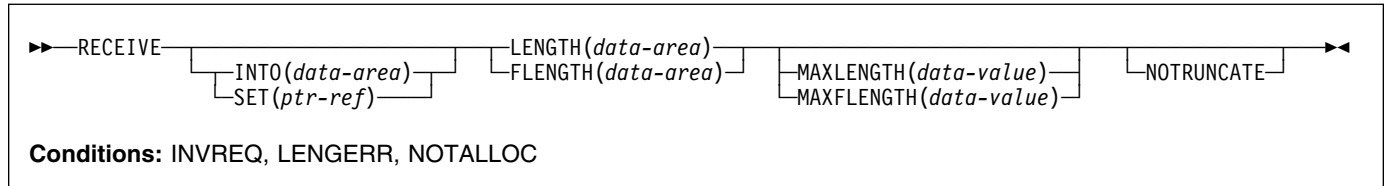
```
EXEC CICS READQ TS
      ITEM(1)
      QUEUE(UNIQNAME)
      INTO(DATA)
      LENGTH(LDATA)
```

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 default)

Receive data from standard CICS terminal support (BTAM or TCAM) or from a task that is not attached to a terminal.



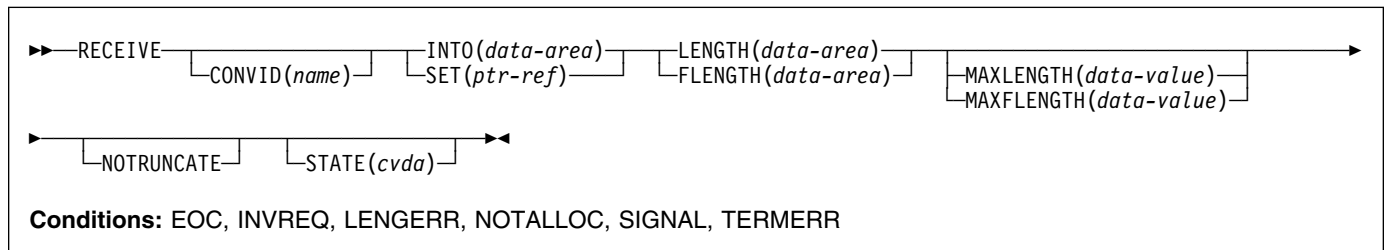
Description

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.



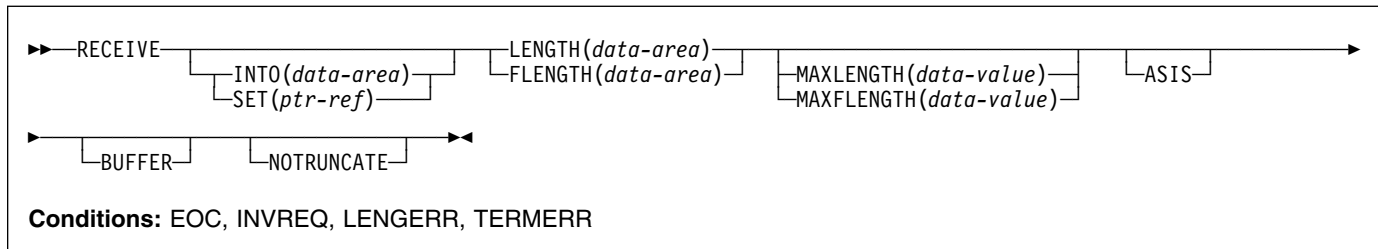
Description

RECEIVE receives data from the conversation partner in an APPC mapped conversation.

RECEIVE (VTAM)

RECEIVE (LUTYPE2/LUTYPE3)

Receive data from a 3270-display logical unit (LUTYPE2) or a 3270-printer logical unit (LUTYPE3).



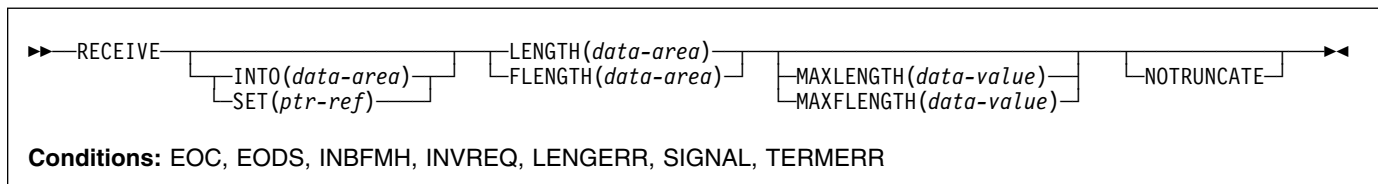
Description

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.



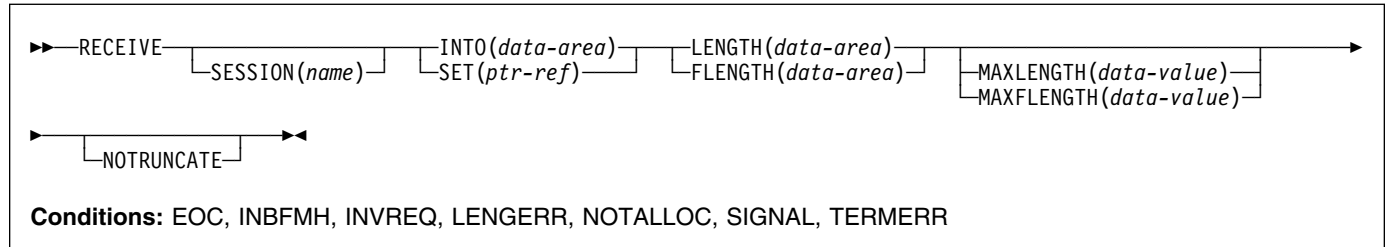
Description

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 (LUTYPE6.1)

Receive data on an LUTYPE6.1 session.

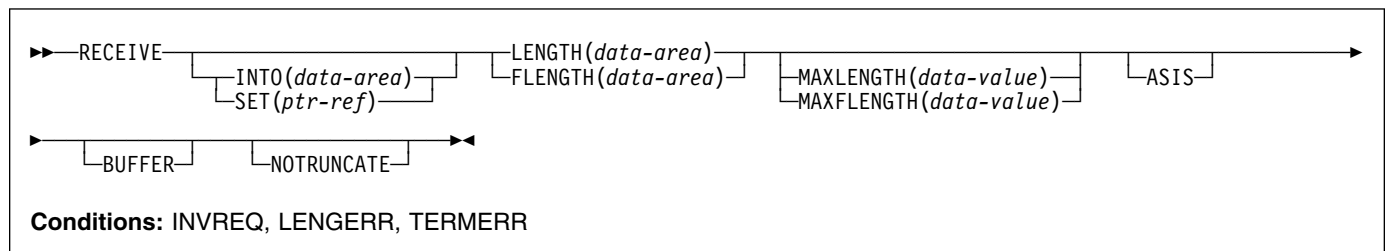


Description

RECEIVE receives data from the conversation partner in an LUTYPE6.1 conversation.

RECEIVE (3270 logical)

Receive data from a 3270 logical unit.



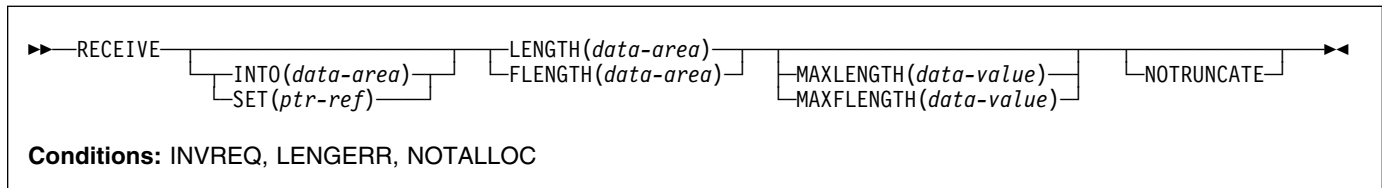
RECEIVE receives data from a terminal.

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 (VTAM)

RECEIVE (3600 pipeline)

Receive initial input data from a 3600 pipeline logical unit. Subsequent RECEIVES for further input data are not allowed.

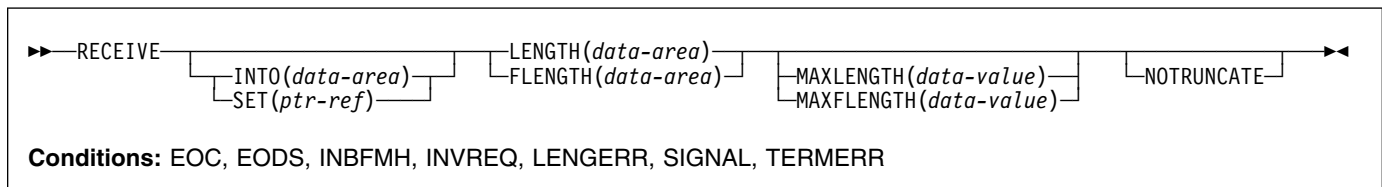


Description

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.



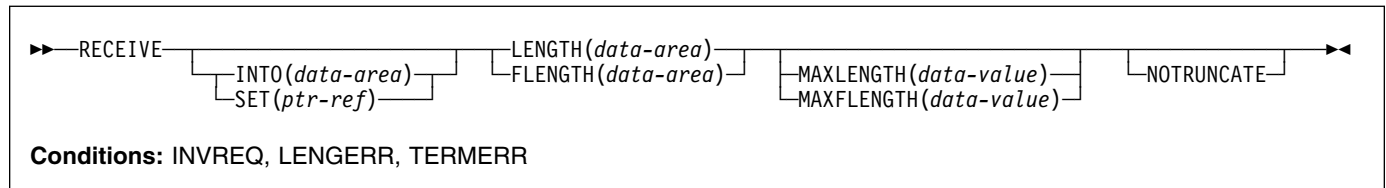
Description

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 (3600-3614)

Receive data from a 3600 (3614) logical unit.



Description

RECEIVE receives data from the terminal.

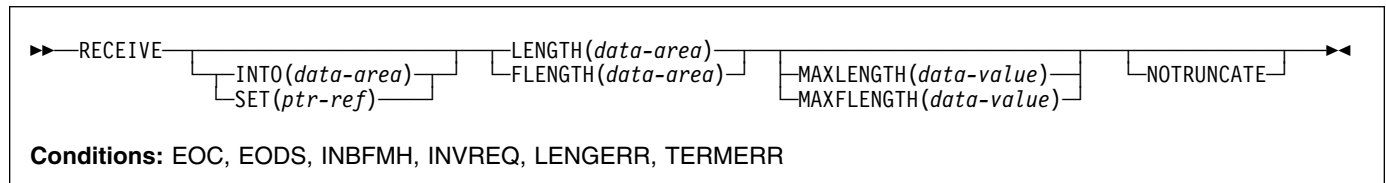
The data-stream and communication format used between a CICS application program and a 3614 is determined by the 3614. The application program is therefore device dependent when handling 3614 communication.

For further information about designing 3614 application programs for CICS, refer to the *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:

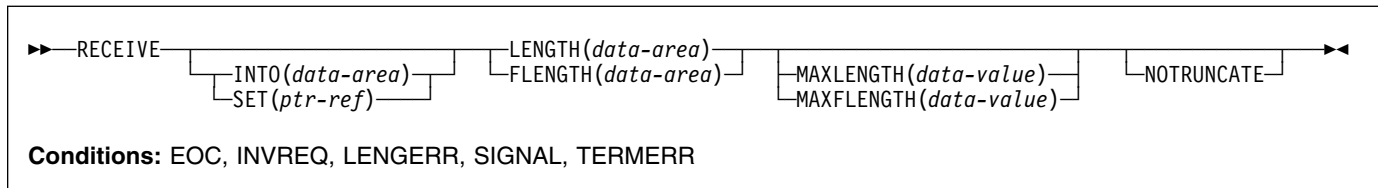
- Interpreter logical unit
- Host conversational (3270) logical unit
- Host conversational (3653) logical unit
- 3650/3680 command processor logical unit

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 (VTAM)

RECEIVE (3767)

Receive data from a 3767 interactive logical unit.



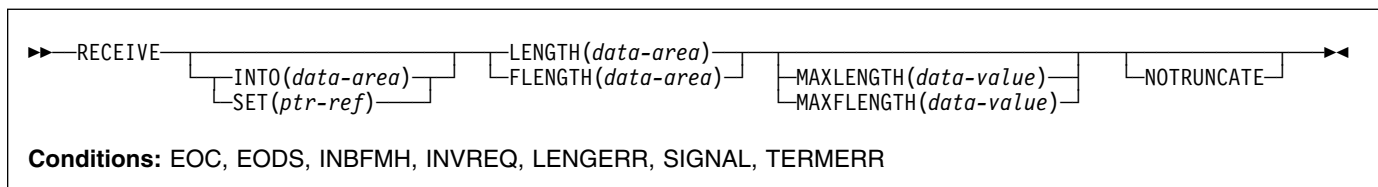
Description

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 (3770)

Receive data from a 3770 batch logical unit.



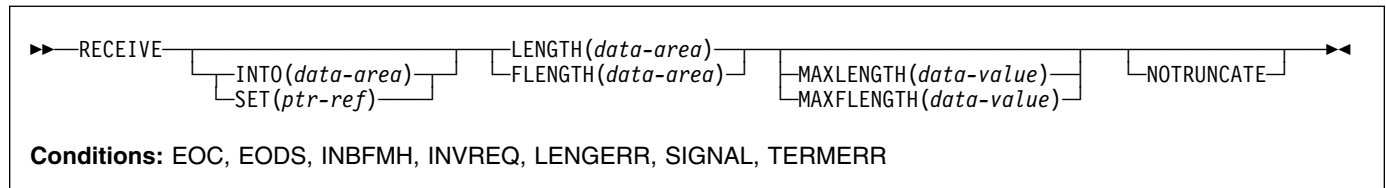
Description

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.

RECEIVE (VTAM)

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 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
 CONFREE
 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.

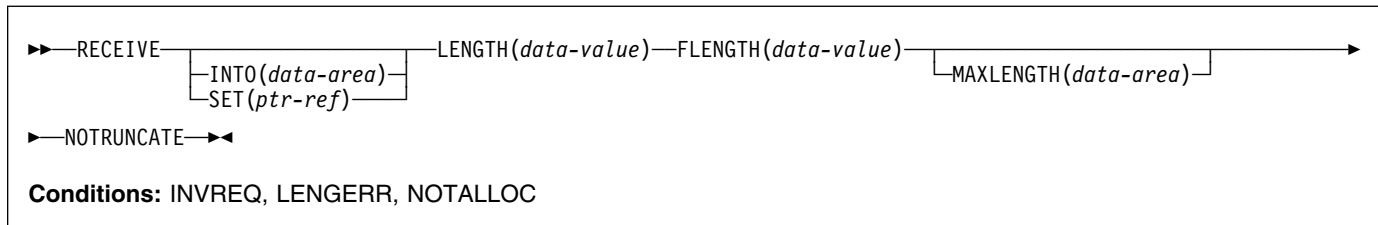
A CANCEL TASK request by a user node error program (NEP) may cause this condition if the task has an outstanding terminal control request active when the node abnormal condition program handles the session error.

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

RECEIVE (non-VTAM)

RECEIVE (non-VTAM default)

Receive data from standard CICS terminal support (BTAM or TCAM) or from a task that is not attached to a terminal.



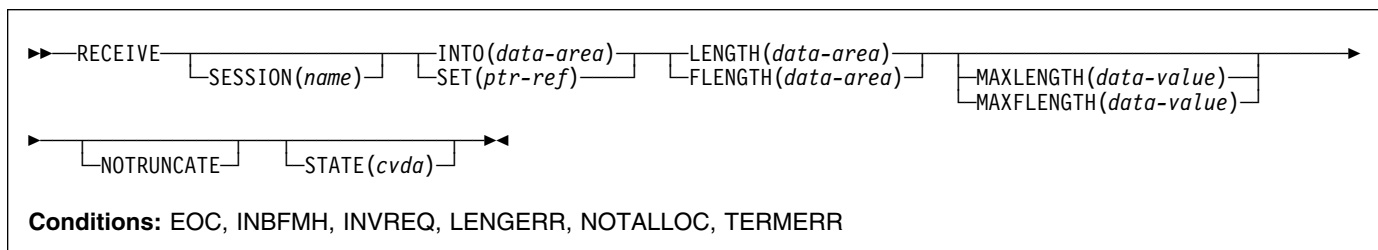
Description

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.

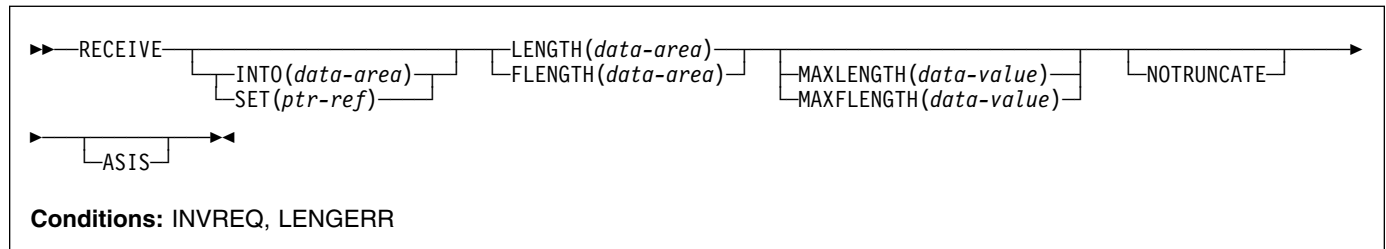


Description

RECEIVE receives data from the conversation partner in an MRO conversation.

RECEIVE (System/3)

Receive data from a System/3.



Description

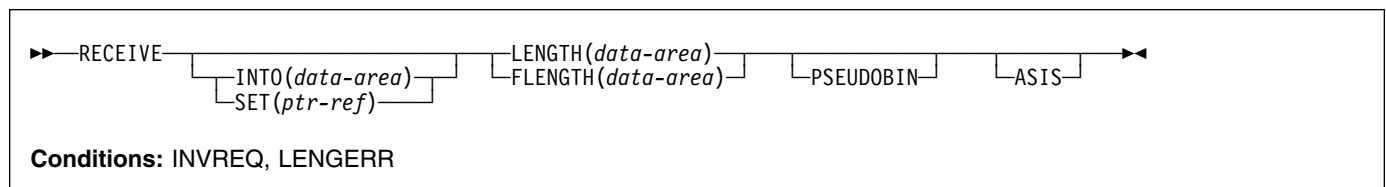
RECEIVE receives data from the terminal. This form of the RECEIVE command also applies to the following devices:

- 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

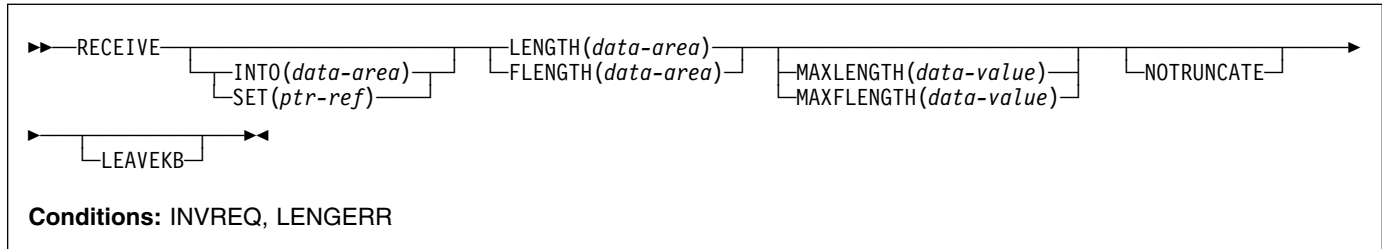
RECEIVE receives data from the terminal. For a description of running with a System/7, see “CONVERSE (System/7)” on page 54.

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 (2260)

Receive data from a 2260 or 2265 display station.



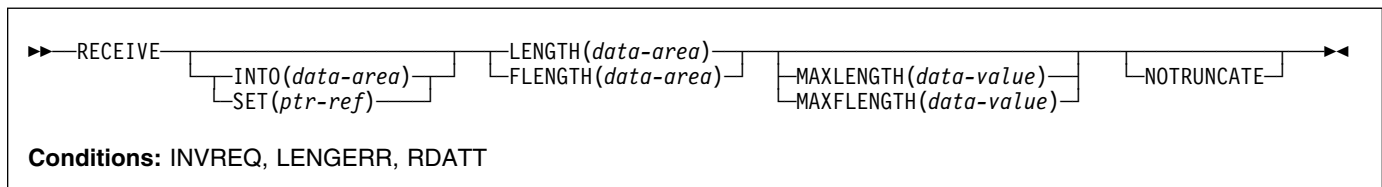
Description

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.



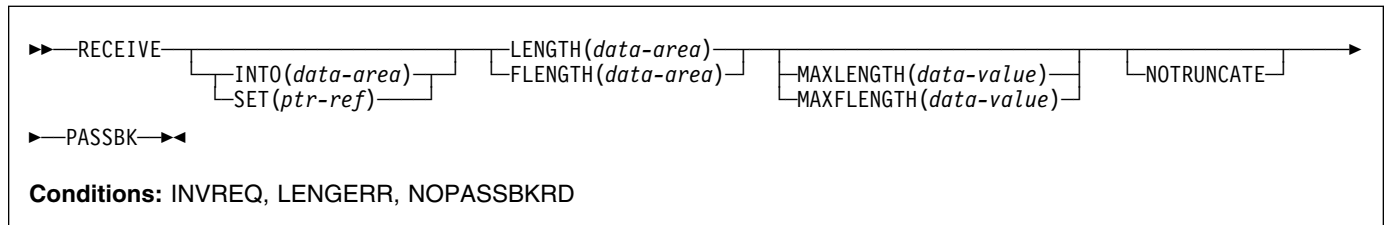
Description

RECEIVE receives data from the terminal. For more information about using a 2741 communication terminal, see “CONVERSE (2741)” on page 56.

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 (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.

RECEIVE (non-VTAM)

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-n-A or STATION-n-N-  
#  
# STATION_n_A or STATION_n_N
```

where **n** is an integer (0 through 9) and A and N signify alternate and 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.

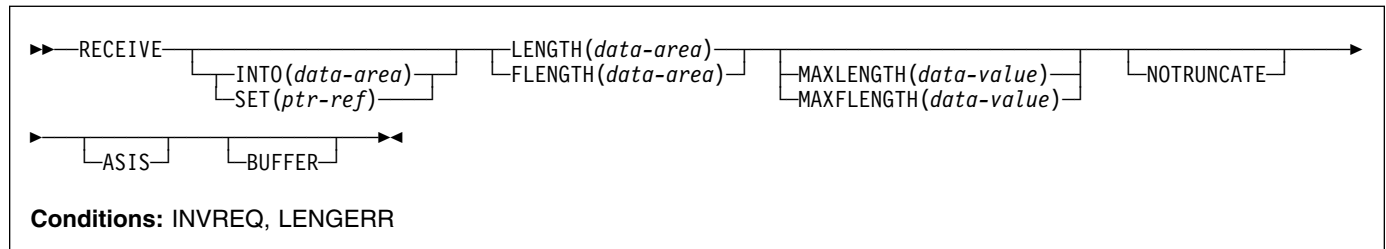
Any of the DFH2980 values TAB-ZERO through TAB-NINE for COBOL and PL/I, may be compared with NUMTAB to find out the number of tab characters that need to be inserted in an output message to get correct positioning of the print element. The tab character is included in DFH2980 as TABCHAR.

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 (3270 display)

Receive data from a 3270 information display system (BTAM or TCAM).



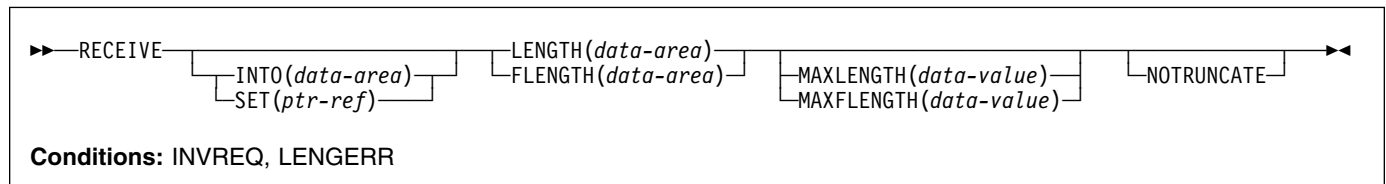
Description

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 58 and "BTAM programmable terminals" on page 394.



Description

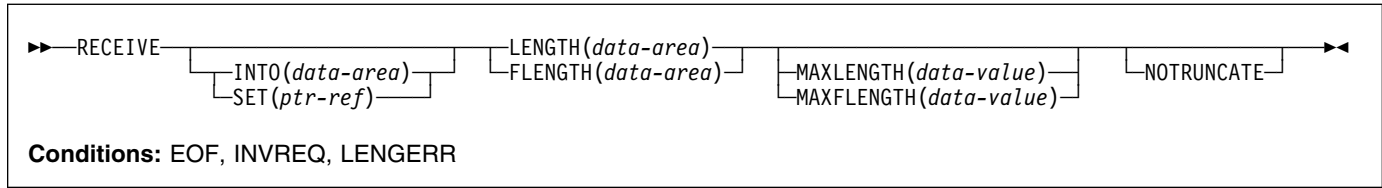
RECEIVE receives data from the terminal. For more information about using the 3600 or 4700 finance communication system, see "CONVERSE (3600 BTAM)" on page 58.

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 (3735)

Receive data from a 3735 Programmable Buffered Terminal.



Description

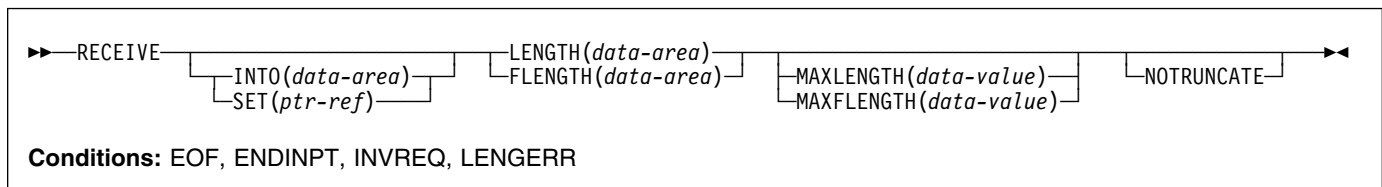
RECEIVE receives data from the terminal. The 3735 Programmable Buffered Terminal may be serviced by CICS in response to terminal-initiated input (automatic answering), or as a result of an automatic (automatic calling) or time-initiated transaction.

For more information about using the 3735 terminal, see “CONVERSE (3735)” on page 60.

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

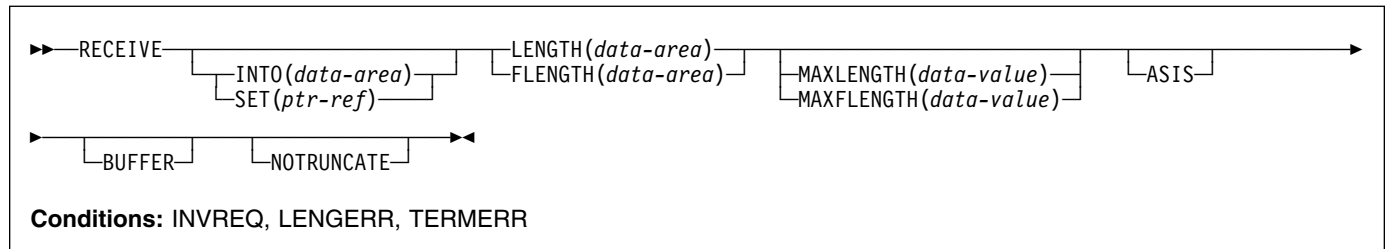
RECEIVE receives data from the terminal.

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 (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.

RECEIVE (non-VTAM)

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

(not TCAM) 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.

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 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.

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 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
FREE
PENDFREE
RECEIVE
ROLLBACK
SEND
SYNCFREE
SYNCRECEIVE
SYNCSEND

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 (not TCAM)

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 (not TCAM)

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 (not TCAM)

occurs if a RECEIVE command is terminated by the attention (ATTN) key rather than the return key.

Default action: ignore the condition.

TERMERR

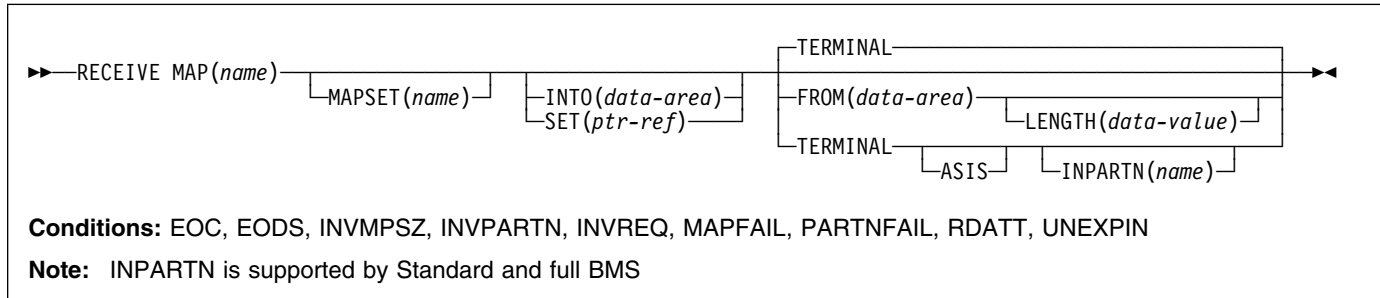
occurs for a terminal-related error, such as a session failure. This condition applies to VTAM-connected terminals only.

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 417 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 431 and 438).

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 431 and 438).

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 associated with the application program, the address of the data may 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.

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.

INVMPsz

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.

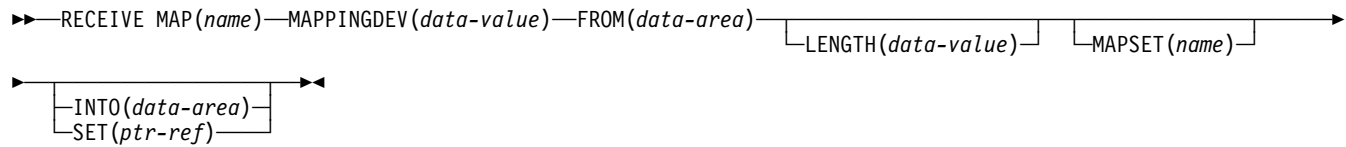
UNEXPIN

occurs when unexpected or unrecognized data is received. This only applies to batch data interchange terminals.

Default action: terminate the task abnormally.

RECEIVE MAP MAPPINGDEV

Receive screen input into an application data area, without reference to the principal facility, if any. Terminal characteristics are obtained from the MAPPINGDEV parameter. For further information about BMS, see the *CICS Application Programming Guide*.



Conditions: INVMPSTZ, INVREQ, MAPFAIL,

Description

RECEIVE MAP MAPPINGDEV allows the mapping of input data from a 3270 terminal that is not necessarily the principal facility of the transaction.

MAPPINGDEV specifies the name of a 3270 terminal whose BMS characteristics were used to create the input data stream. This may be a terminal from which the data was originally received using a RECEIVE command.

Options

AID(*data-value*)

specifies the one-byte data area containing the value of the 3270 attention identifier (AID) to be used when performing the mapping operation. Usually this will be the value contained in EIBAID following the RECEIVE operation that originally received the datastream from the terminal.

The value specified is moved into field EIBAID in the EXEC interface block on completion of the operation. No check is made that the AID value specified is valid.

If AID(*data-value*) is not specified, then the AID value defaults to X'7D' (the Enter key).

If the AID byte (either explicitly, or by default) indicates an operation other than CLEAR, PA1, PA2, or PA3, and CURSLOC=YES is specified for the map, then the field containing the cursor is flagged by setting the X'02' bit in its flag byte.

If the AID (whether specified explicitly, or by default) is the subject of a HANDLE AID command, the specified branch will be taken in the usual way.

CURSLOC(*data-value*)

specifies an unsigned halfword binary field containing the cursor position (relative to zero) to be used. Usually this will be the value contained in EIBCPOSN following the RECEIVE operation that originally received the datastream from the terminal.

The value specified is moved into EIBCPOSN in the EXEC interface block on completion of the operation. No check is made that the CURSLOC value specified is valid.

If CURSLOC(*data-value*) is not specified, then the cursor value defaults to X'0000'.

FROM(*data-area*)

specifies the data area containing the data to be mapped. This must be in the format of a TIOA and must contain a 12-byte prefix.

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 431 and 438).

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.

MAPPINGDEV(*data-value*)

specifies the name of a 3270 terminal whose characteristics match those of the terminal from which the data was originally received using a RECEIVE command.

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 or RECEIVE MAP command, or until the end of the transaction, unless FREEMAINed by the application.

| If "TASKDATALOC(ANY)" is specified for the running task, the data returned may be above or below the 16MB line.

| If "TASKDATALOC(BELOW)" is specified for the running task, the data returned is below the 16MB line.

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

Conditions

| Some of the following conditions may occur in combination. If more than one occurs, only the first is passed to the application program.

INVMPSZ

| occurs if the specified map is too wide or too long for the terminal.

| Default action: terminate the task abnormally.

INVREQ

| occurs if the terminal specified by MAPPINGDEV does not exist, does not support BMS, or is not a 3270 printer or display.

| 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.

| 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 417 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.

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 associated with the application program, the address of the data may 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.

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.

Default action: truncate the data to fit within the INTO area.

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.

Note: RELEASE does not remove a program from storage. It reduces the RESCOUNT by 1 and when the count reaches zero, the storage occupied by the program can be released by CICS storage manager.

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.

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

RELEASE

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 Either the command is issued for a program that was loaded, without the HOLD option, by another task; or the program has been enabled as a global user exit .
- #
- #
- 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.
- 42 An attempt has been made to RELEASE a JVM program. This is invalid because Java byte codes programs are not managed by CICS Loader.

Default action: terminate the task abnormally.

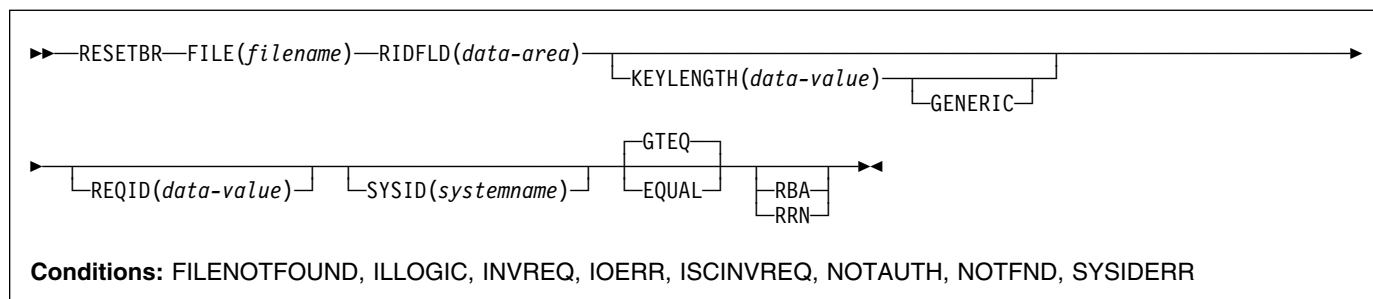
Examples

The following example shows how to release an application program, called PROG4, loaded in response to a LOAD command:

```
EXEC CICS RELEASE PROGRAM('PROG4')
```

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 BDAM 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 BDAM 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 or RLS 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.

RESETBR

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
- Coupling facility data tables
- KSDS or ESDS files that hold more than 4GB
- Any KSDS files opened in RLS access mode

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 BDAM 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 377.)

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.
- 44 The command does not conform to the format of RESETBR for a user-maintained or coupling facility data table; for example, RBA is specified.
- 51 A RESETBR command to a KSDS file that is being accessed in RLS mode specifies the RBA keyword. RLS mode does not support RBA access to KSDS data sets.

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 377.)

For a coupling facility data table, an IOERR indicates a bad response returned from a coupling facility access.

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.

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

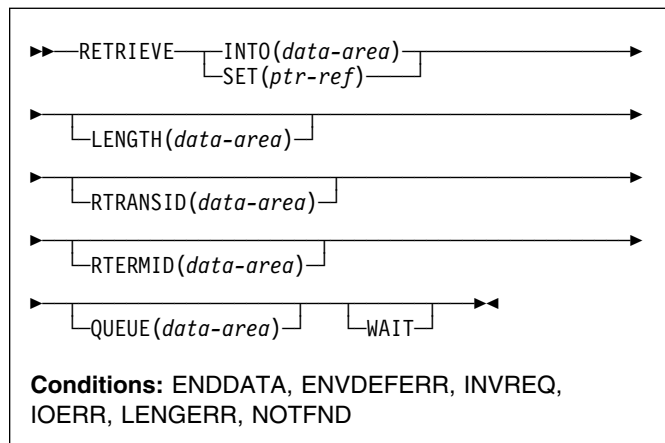
RESP2 values:

- 130** The SYSID option specifies a name that is neither the local CICS region nor a remote system (as defined by a CONNECTION definition). SYSIDERR also occurs when the link to the remote system is closed
- 131** For a coupling facility data table, the connection to the coupling facility data table server has failed. This could be because the server itself has failed, or the server is available, but CICS has failed to connect to it.
- 132** The RESETBR is issued against a coupling facility data table that no longer exists, probably because of a coupling facility failure, in which case the coupling facility data table server also fails. See the *CICS System Definition Guide* for information about restarting a coupling facility data table server and reloading a table.

Default action: terminate the task abnormally.

RETRIEVE

Retrieve data stored for a task.



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 normally released on execution of the RETRIEVE command, or on termination of the task if no RETRIEVE command is executed prior to termination.

If the START command specified ATTACH, the storage is not released. (ASSIGN STARTCODE in such a task returns 'U' rather than 'S' or 'SD').

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.

RETRIEVE

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.

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 associated with the application program, the address of the data may 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.

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 DEFINE TRANSACTION.)

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 TERMID 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)
```

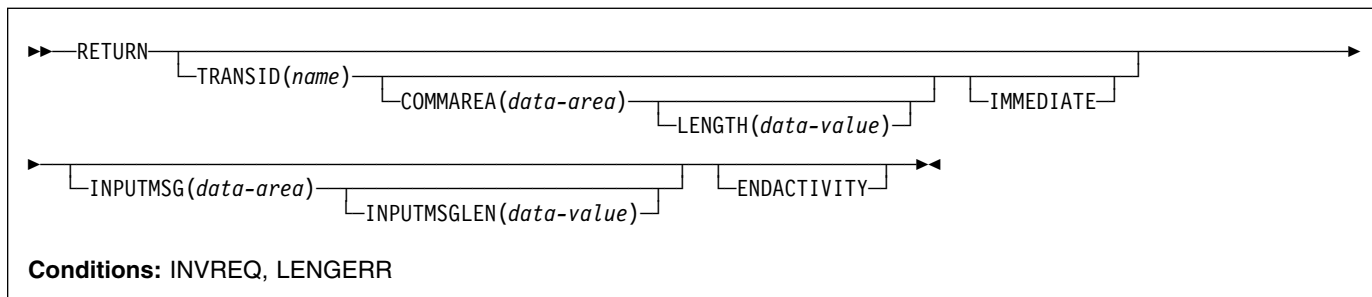
The following example shows how to request retrieval of a data record stored for a task into a data area provided by CICS; the pointer reference (PREF) specified by the SET option is set to the address of the storage area reserved for the data record.

```
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.

For information about the use of this command in the CICS BTS environment, see the *CICS Business Transaction Services* manual.

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.

ENDACTIVITY

This option is for use by programs that implement CICS business transaction services (BTS) activities.

For information about BTS in general and the ENDACTIVITY option in particular, see the *CICS Business Transaction Services* manual.

This option is ignored outside the CICS BTS environment.

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.

- 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.
- 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:

REWIND COUNTER

Rewind a named counter that has reached its limit (that is, the maximum number has been assigned).

```
► REWIND COUNTER(name) [ POOL(name) ] [ INCREMENT(data-value) ] ◄
```

Conditions: INVREQ, SUPPRESSED

```
► REWIND DCOUNTER(name) [ POOL(name) ] [ INCREMENT(data-area) ] ◄
```

Conditions: INVREQ, SUPPRESSED

Description

These counter commands reset the current value of the named counter to its defined minimum number. COUNTER operates on fullword counters and DCOUNTER operates on doubleword counters.

For information about specifying fullword and doubleword variables on these named counter commands, see “Argument values” on page 2.

Options

COUNTER(*name*)

specifies the name of the named counter that is to be reset to its minimum value. The name can be up to 16 alphanumeric characters. If *name* is a variable that contains a name that is less than 16 characters, the name must be padded with trailing blanks.

DCOUNTER(*name*)

specifies the name of the named counter that is to be reset to its minimum value. The name can be up to 16 alphanumeric characters. If *name* is a variable that contains a name that is less than 16 characters, the name must be padded with trailing blanks.

INCREMENT(*data-value*)

specifies, as a fullword signed binary value (doubleword unsigned binary value for DCOUNTER), an increment to be used in determining whether the named counter is in a valid state to be reset. If a previous GET command (which did not specify the REDUCE option) specified an increment that caused the GET command to fail, specify the same increment on the REWIND. The named counter server applies the increment before testing whether the counter is in a counter-at-limit condition.

See the the INCREMENT option on the GET command for more details.

POOL(*poolname*)

specifies an 8-character string to be used as a pool selection parameter to select the pool in which the named counter resides. The string can be a logical pool name, or the actual pool name.

Valid characters for the pool selector string are A through Z, 0 through 9, \$ @ # and _ (underscore). If *name* is a variable that contains a name that is less than 8 characters, the name must be padded with trailing blanks.

This parameter is optional. If you omit the name of the pool, a pool selector value of 8 blanks is assumed.

If there is no matching entry in the DFHNCOPT options table, CICS uses the name specified on the NCPLDFT system initialization parameter, which specifies the default named counter pool.

For information about generating a named counter options table using the DFHNCO macro, see the *CICS System Definition Guide*.

Conditions

INVREQ

RESP2 values:

201 Named counter not found.

301 The server has reported an error code that is not understood by the named counter interface. Generally, this is not possible unless the interface load module, DFHNCF, is at a lower maintenance or release level than the server itself.

303 An unexpected error, such as structure failure or loss of connectivity, has occurred on a macro used to access the coupling facility. Further information can be found in message DFHNC0441 in the CICS job log.

- | **304** The pool selection parameter specified in the
| program cannot be resolved to a valid server
| name using the current options table.
- | **305** The interface is unable to establish a connection
| to the server for the selected named counter pool.
| Further information can be found in an AXM
| services message (AXMSC $nnnn$) in the CICS job
| log.
- | **306** An abend occurred during server processing of a
| request. Further information can be found in a
| message in the CICS job log and the server job
| log.
- | **308** The DFHNCOPT options table module, required
| for resolving a pool name, could not be loaded.
- | **309** During processing of the options table, the named
| counter interface encountered an unknown entry
| format. Either the options table is not correctly
| generated, or the DFHNCIF interface load module
| is not at the same release level as the options
| table.
- | **310** An options table entry matching the given pool
| name specified a user exit program, but the user
| exit program is not link-edited with the options
| table and cannot be loaded.
- | **403** The POOL parameter contains invalid characters
| or embedded spaces.
- | **404** The COUNTER parameter contains invalid
| characters or embedded spaces.
- | **406** The INCREMENT value is invalid. The value
| specified cannot be greater than the total range of
| the counter ((maximum value – minimum value)
| + 1).

| Default action: terminate the task abnormally.

| **SUPPRESSED**

| RESP2 values:

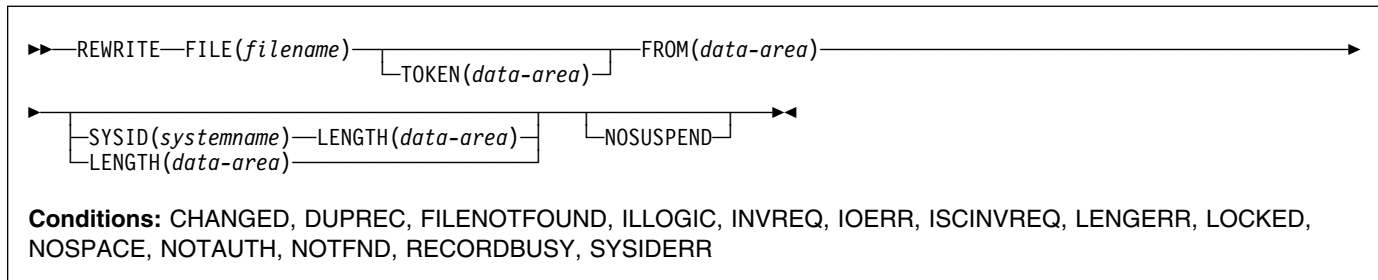
- | **102** The named counter has not yet reached its limit
| (that is, the current value is not equal to the
| maximum value plus 1, giving the counter-at-limit
| condition). This error condition is returned if the
| named counter is not at its limit even after
| applying any specified increment.

| Default action: terminate the task abnormally.

REWRITE

REWRITE

Update a record in a file.



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.

When this command is used to update records in a coupling facility data table, the update is made only to the data table in the coupling facility.

Note: The only data sets greater than 4GB supported by
CICS are KSDS, and then only if they are accessed
by key. CICS does not support ESDS or RRDS data
sets defined with the extended attribute.

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-area)

specifies the length, as a halfword binary value, of the data area where the record is written from.

This option must be specified if SYSID is specified.

If the file is on a remote system, the LENGTH parameter need not be set here but must be set in the file resource definition.

If the file is on a local system, the LENGTH parameter must be set for variable-length records, using the INTO option, but not for fixed-length records. It is, however, advisable to specify the length for fixed-length records because it causes a check to be made that the record being written is not longer than that defined for the data set.

NOSUSPEND (RLS only)

The request does not wait if VSAM is holding an active lock against the record, including records locked as the result of a DEADLOCK.

Generally, you should not need this option because, for example, when re-writing a record to a base-only data set, the active lock was acquired when the task issued the READ UPDATE.

However, lock contention can occur if the update involves changes made in RLS mode to records in a VSAM data set that has one or more alternate indexes, and an alternate index is defined with unique keys.

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.

TOKEN can be function shipped. However, if a request specifying TOKEN is function shipped to a member of the CICS family of products that does not support the TOKEN option, the request fails:

Conditions

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

CHANGED

RESP2 values:

109 A REWRITE command is issued for a file that is defined as a coupling facility data table using the contention update model and the record has been changed since the application program read it for update. To successfully update the record, repeat the read for update to get the latest version of the record, re-apply the change, and try the rewrite again.

Default action: terminate the task abnormally.

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 377 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.

A possible reason for the previous READ for UPDATE not being found is that it failed for some reason, and the failure has not been correctly handled or has been ignored.

46 A REWRITE command has attempted to change the length of a BDAM 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.

55 NOSUSPEND is not allowed because the file is not a VSAM file that is accessed in RLS mode.

56 An attempt to update a recoverable coupling facility data table has failed because the current unit of work has already updated 1024 recoverable coupling facility data tables. You cannot update more than 1024 recoverable coupling facility data tables within a unit of work

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 377 for details.)

For a coupling facility data table, an IOERR indicates a bad response returned from a coupling facility access.

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 The LENGTH option is not specified for a file with variable-length records, or for a BDAM 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

REWRITE

RIDFLD) after it was read for update. This may be caused by a logic error in the application program.

This condition can also occur when a REWRITE command is issued to a coupling facility data table using the contention model and the record has been deleted since it was read for update.

Default action: terminate the task abnormally.

LOCKED

RESP2 values:

- 106** Attempt has been made to update a record, but a retained lock exists against a unique alternate key that is involved in the request.

Default action: abend the task with code AEX8.

NOSPACE

RESP2 values:

- 100** No space is available on the direct access device for adding the updated record to the data set.
- 102** The maximum number of records specified for a recoverable coupling facility data table has been exceeded. This can occur on a rewrite operation because an extra record is required in the coupling facility data table for recovery purposes until the update has been committed.
- 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.
- 108** For coupling facility data tables, this condition occurs if there is insufficient space in the coupling facility data table pool to store the updated record.

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.

RECORDBUSY

RESP2 values:

- 107** NOSUSPEND is specified but VSAM holds an active lock against a unique alternate index key that is involved in the request, which would cause the request to wait. (See the note about active and retained locks below.)

Default action: abend the task with code AEX9.

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.
- 131** For a coupling facility data table, the connection to the coupling facility data table server has failed. This could be because the server itself has failed, or the server is available, but CICS has failed to connect to it.
- 132** The REWRITE is issued against a coupling facility data table that no longer exists, probably because of a coupling facility failure, in which case the coupling facility data table server also fails. See the *CICS System Definition Guide* for information about restarting a coupling facility data table server and reloading a table.

Default action: terminate the task abnormally.

Note: RECORDBUSY refers to active locks and LOCKED refers to retained locks:

- REWRITE requests for records that have *retained* locks are always rejected with a LOCKED response.
- REWRITE requests for records that have *active* locks wait for the lock to be released, except when the NOSUSPEND keyword is specified, in which case CICS returns the RECORDBUSY response.

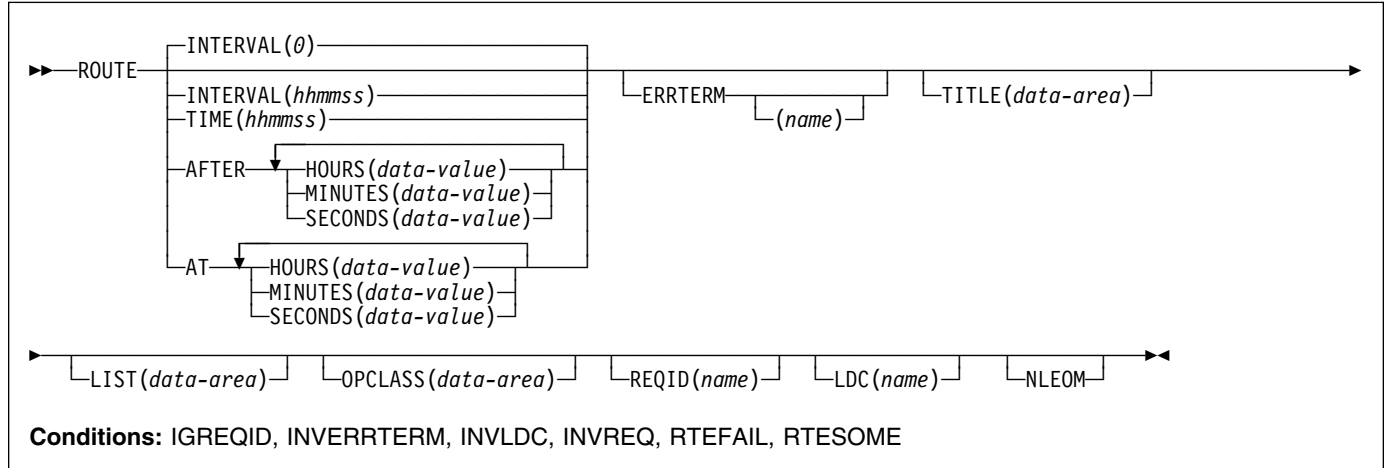
Examples

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.

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 the 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

ROUTE

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 433 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 by the RDO TYPETERM options PAGESIZE or 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 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 0–359999 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.

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.

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

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 (**).

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.

SEND (VTAM)

SEND (VTAM default)

Write data to a standard CICS supported terminal (BTAM or TCAM).

▶▶ SEND FROM(*data-area*) LENGTH(*data-value*)
FLENGTH(*data-value*) DEST(*name*) 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*) ▶▶

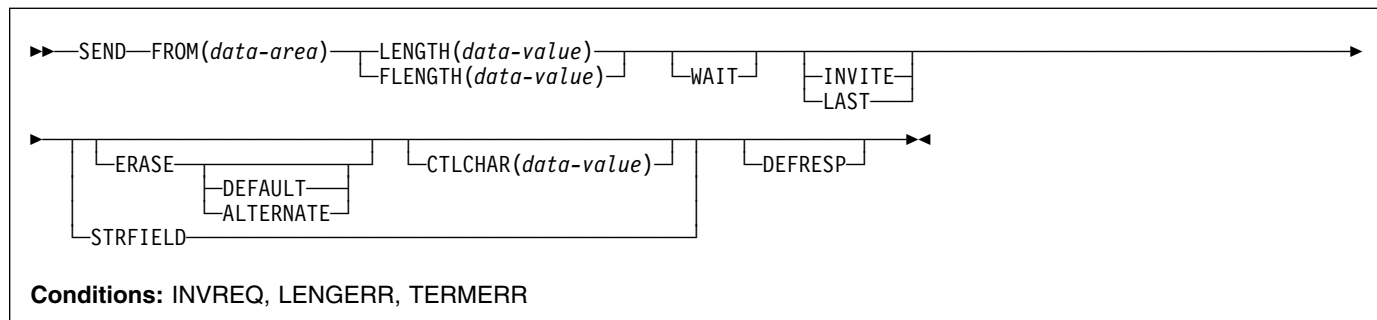
Conditions: INVREQ, LENGERR, NOTALLOC, SIGNAL, TERMERR

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).

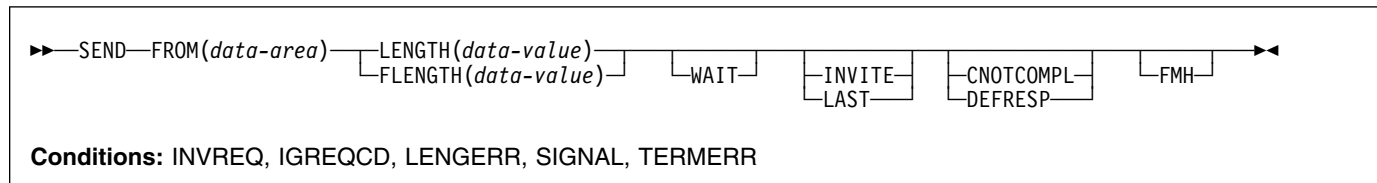


Description

SEND writes data to a terminal.

SEND (LUTYPE4)

Write data to a LUTYPE4 logical unit.



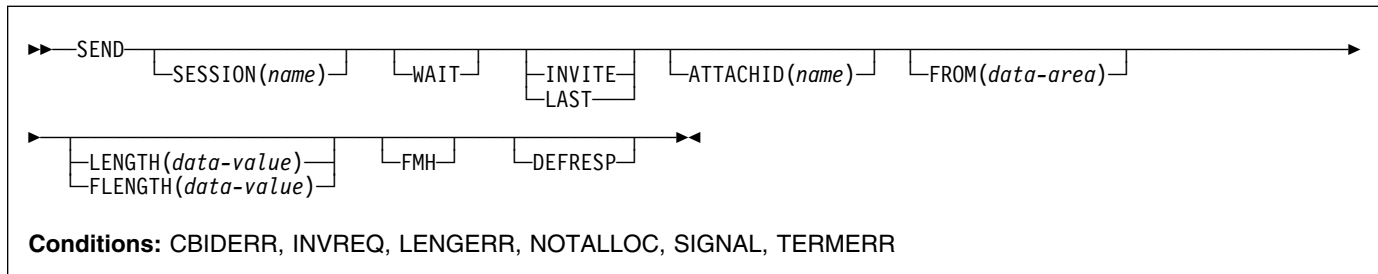
Description

SEND writes data to a terminal.

SEND (VTAM)

SEND (LUTYPE6.1)

Send data on an LUTYPE6.1 conversation.

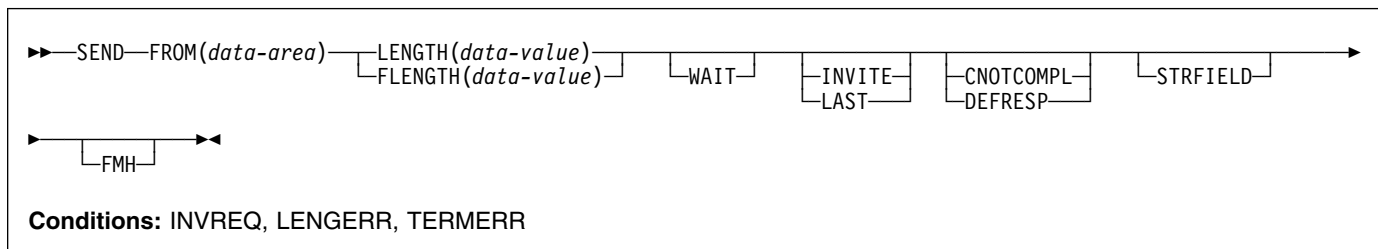


Description

SEND sends data to a conversation partner on an LUTYPE6.1 conversation.

SEND (SCS)

Write data to a 3270 SCS printer logical unit.

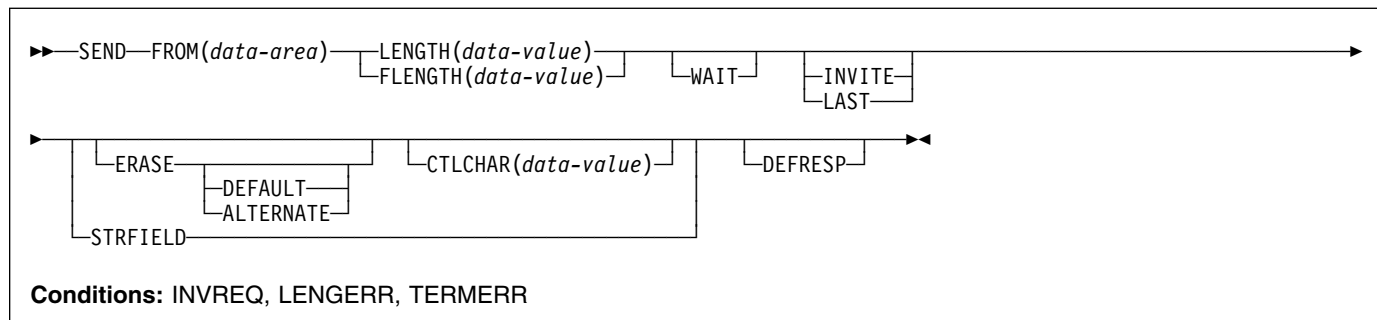


Description

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.

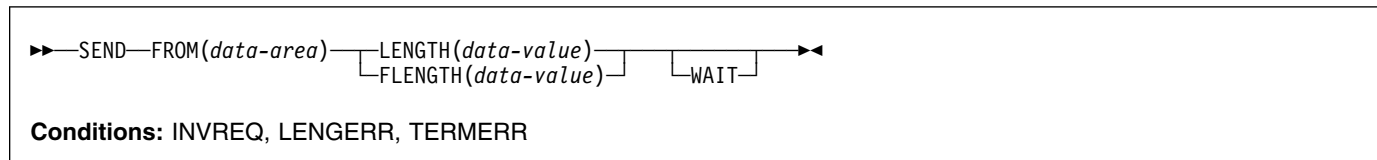


Description

SEND writes data to a terminal.

SEND (3600 pipeline)

Write data to a 3600 pipeline logical unit.



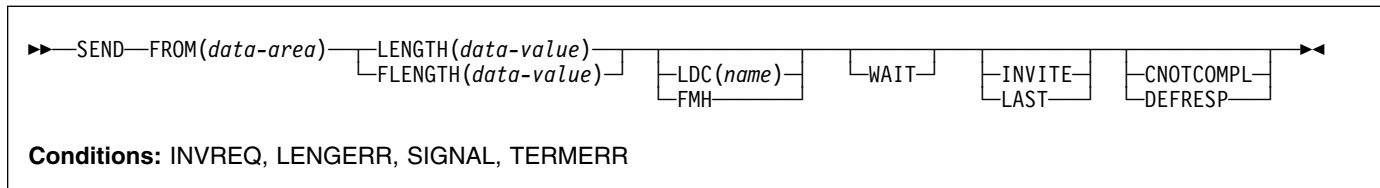
Description

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 4770 and 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).

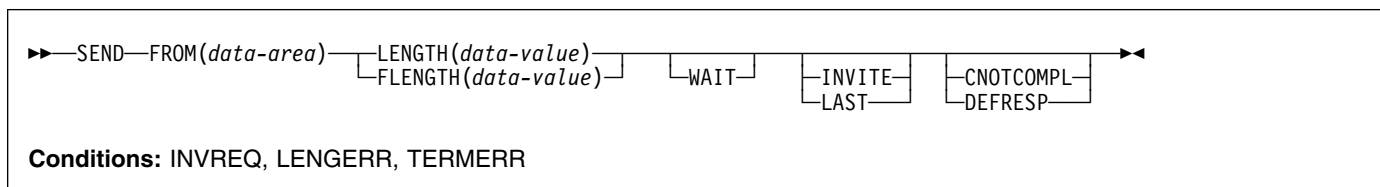
A 3600 device and a logical page size are also associated with an LDC. "LDC" or "LDC value" is used in this book in reference to the code specified by the user. "LDC mnemonic" refers to the 2-character symbol that represents the LDC numeric value.

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.

On output, the FMH can be built by the application program or by CICS. If your program supplies the FMH, you place it at the # front of your output data and specify the FMH option on your SEND command. If you omit the FMH option, CICS will provide # an FMH but you must reserve the first three bytes of the message for CICS to fill in.

SEND (3600-3614)

Write data to a 3600 (3614) logical unit.



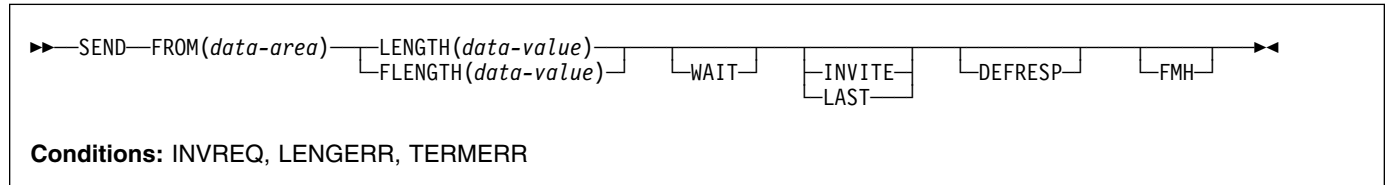
Description

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 *IBM 4700/3600/3630 Guide*.

SEND (3650 interpreter)

Write data to a 3650 interpreter logical unit.

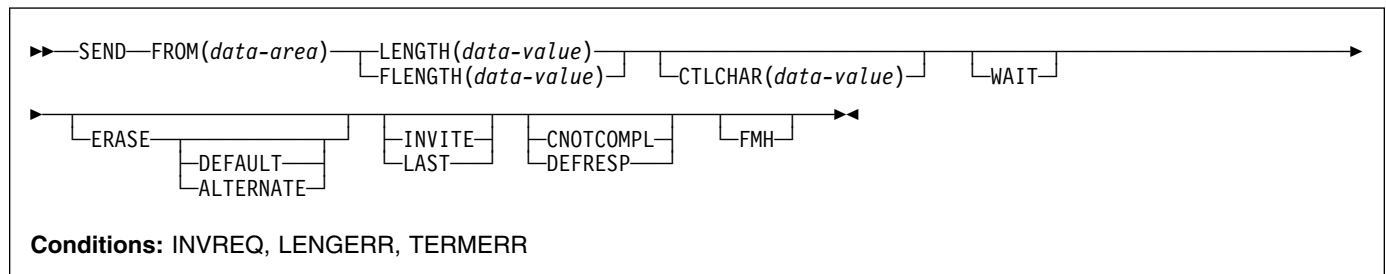


Description

SEND writes data to a terminal.

SEND (3650-3270)

Write data to a 3650 logical unit.



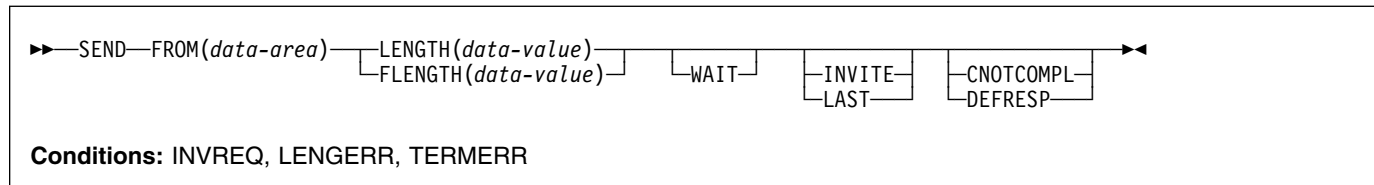
Description

SEND writes data to a terminal.

SEND (VTAM)

SEND (3650-3653)

Write data to a 3650 (3653) logical unit.

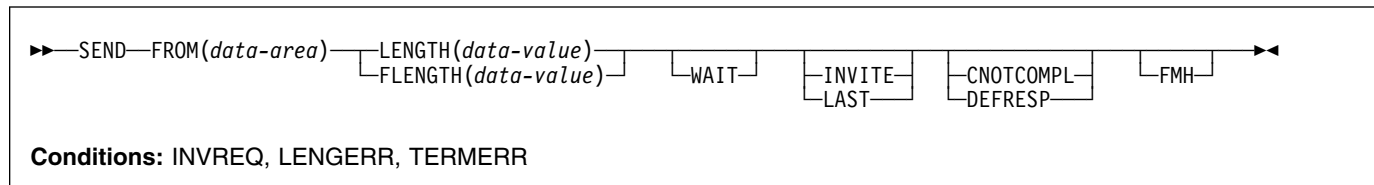


Description

SEND writes data to a terminal.

SEND (3650-3680)

Write data to a 3650 (3680) logical unit.

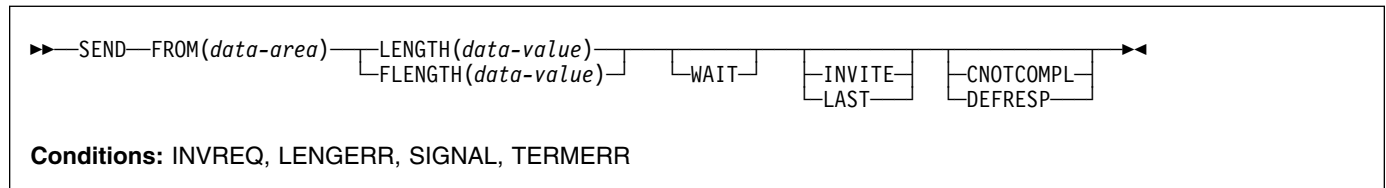


Description

SEND writes data to a terminal.

SEND (3767)

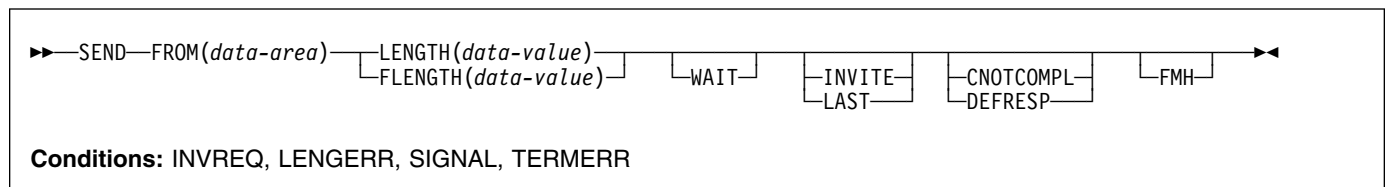
Write data to a 3767 interactive logical unit.

**Description**

SEND writes data to a terminal. This form of SEND also applies to the 3770 interactive logical unit.

SEND (3770)

Write data to a 3770 batch logical unit.

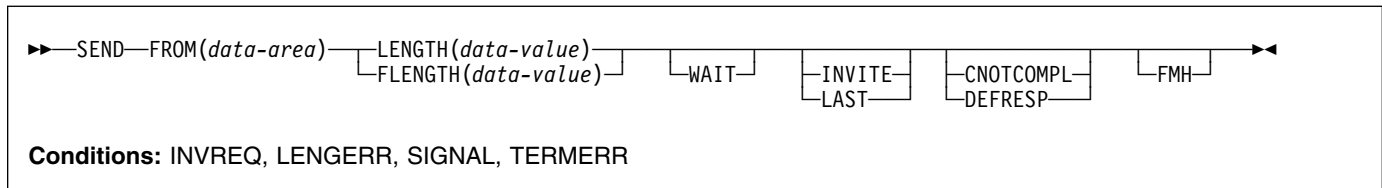
**Description**

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.



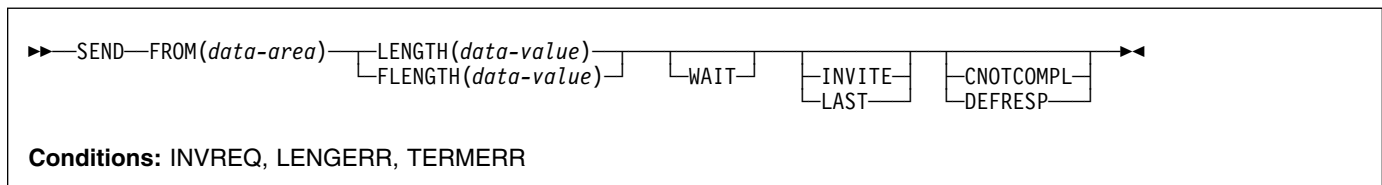
Description

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.

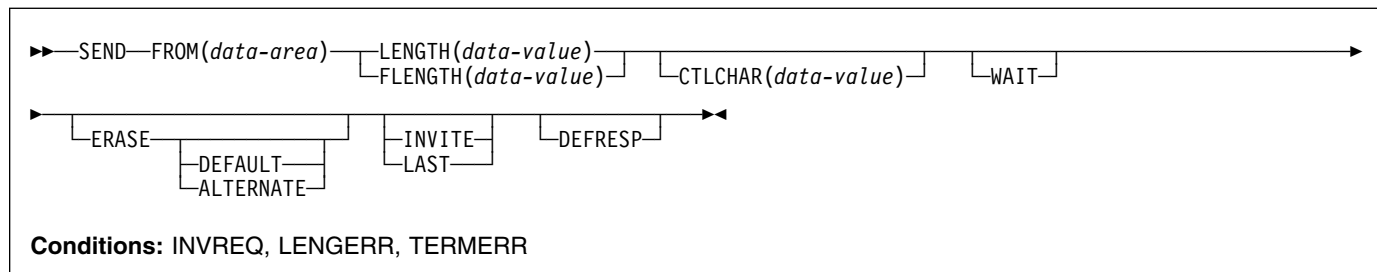


Description

SEND writes data to a terminal.

SEND (3790 3270-display)

Write data to a 3790 (3270-display) logical unit.

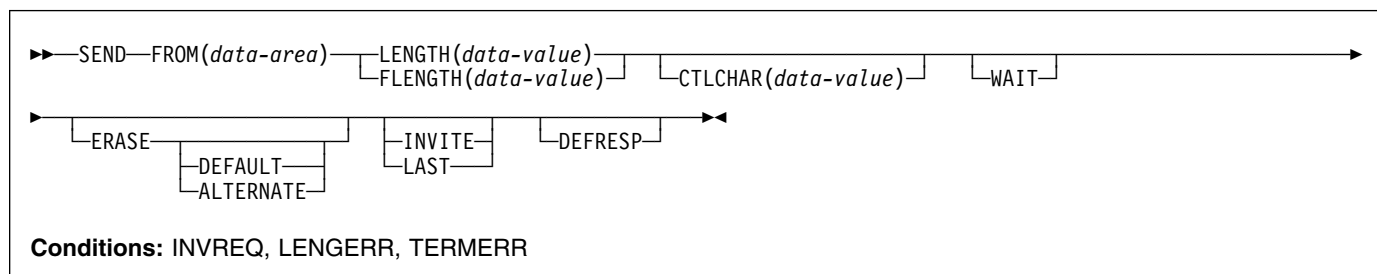


Description

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

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.

DEST(*name*)

specifies the 4-byte symbolic name of the TCAM destination to which the message is to be sent. This

option is meaningful only for terminals defined using DFHTCT TYPE=SDSCI with DEVICE=TCAM.

If you use the DEST option, you must be aware of any restrictions placed on device-dependent data streams by the message control facility in use. See the programming information about the CICS-TCAM interface in the *CICS Customization Guide*.

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.

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.

The use of FMH is optional and is not supported for all
terminal types. If not supplied, CICS takes no action,
except for 3600/4700 terminals, where an FMH is
mandatory. In this case, if FMH is not specified, CICS
supplies one and places it in the first 3 bytes of the
message, which you must reserve for this purpose.

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.

[†] Documented in the *IBM 3270 Data Stream Programmer's Reference*.

LDC(*name*)

specifies the 2-character mnemonic used to determine the appropriate logical device code (LDC) numeric value. The mnemonic represents an LDC entry in the terminal control table TYPE=LDC.

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
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

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 (BTAM or TCAM).

```

▶▶ SEND FROM(data-area) ┌ LENGTH(data-value) ───┐
                        └ FLENGTH(data-value) ─┘  ┌ DEST(name) ─┐
                                                    └ 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 ─┐
                                      └ LAST ─┘  ┌ ATTACHID(name) ─┐
                                                    └ FROM(data-area) ─┘
  ───────────────────────────────────────────────────────────────────────────────────▶
▶ ───────────┐
              └ LENGTH(data-value) ─┐
              └ FLENGTH(data-value) ─┘  ┌ FMH ─┐
                                          └ DEFRESP ─┐
                                                  └ STATE(cvda) ─┘
  
```

Conditions: CBIDERR, INVREQ, LENGERR, NOTALLOC, TERMERR

Description

SEND sends data to a conversation partner on an MRO conversation.

SEND (non-VTAM)

SEND (System/3)

Write data to a System/3.

▶▶SEND FROM(*data-area*) — LENGTH(*data-value*)
FLENGTH(*data-value*) — DEST(*name*) — WAIT — ASIS — CNOTCOMPL ▶▶

Conditions: INVREQ, LENGERR

Description

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*) — DEST(*name*) — WAIT — PSEUDOBIN — ASIS ▶▶

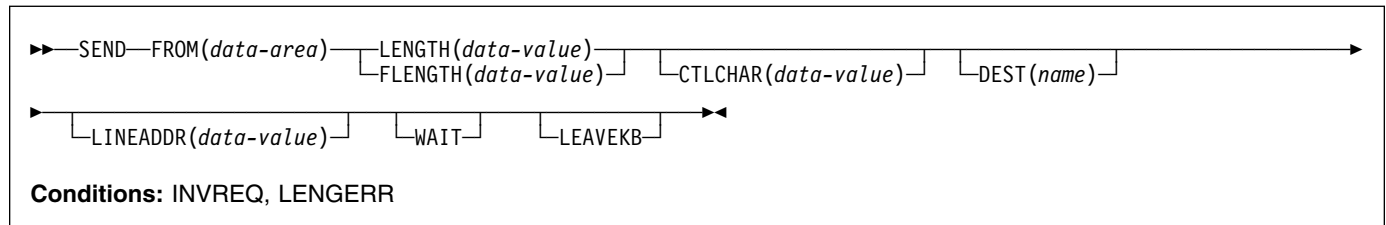
Conditions: INVREQ, LENGERR

Description

SEND writes data to a terminal. For a description of running with a System/7, see “CONVERSE (System/7)” on page 54.

SEND (2260)

Write data to a 2260 or 2265 display station.

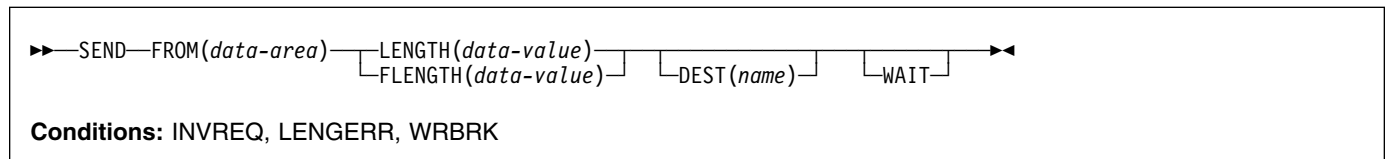


Description

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 56.

SEND (non-VTAM)

SEND (2980)

Write data to a 2980 general banking terminal system.

►► SEND FROM(*data-area*) — LENGTH(*data-value*) — FLENGTH(*data-value*) — DEST(*name*) — PASSBK — 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 233.

SEND (3270 display)

Write data to a 3270 information display system (BTAM or TCAM).

►► SEND FROM(*data-area*) — LENGTH(*data-value*) — FLENGTH(*data-value*) — DEST(*name*) — WAIT — ◄◄

►

ERASE — CTLCHAR(*data-value*) — ◄◄

STRFIELD — DEFAULT — ALTERNATE —

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 394.

```

▶▶SENDFROM(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 58.

SEND (3735)

Write data to a 3735 Programmable Buffered Terminal.

```

▶▶SENDFROM(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 60.

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

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.

DEST(*name*)

specifies the 4-byte symbolic name of the TCAM destination to which the message is to be sent. This option is meaningful only for terminals defined using DFHTCT TYPE=SDSCI with DEVICE=TCAM.

If you use the DEST option, you must be aware of any restrictions placed on device-dependent data streams by the message control facility in use. See the programming information about the CICS-TCAM interface in the *CICS Customization Guide*.

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.

[†] Documented in the *IBM 3270 Data Stream Programmer's Reference*.

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 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
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**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 with abend code ATNI.

WRBRK

occurs if the command is terminated by the attention key.

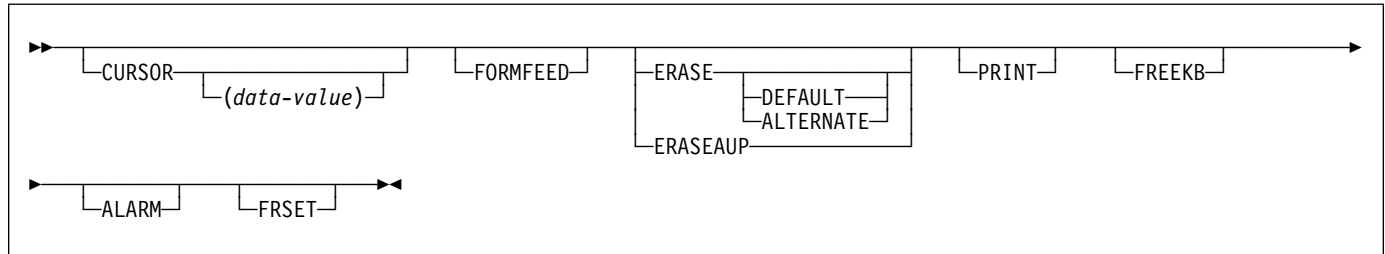
Default action: ignore the condition.

SEND CONTROL

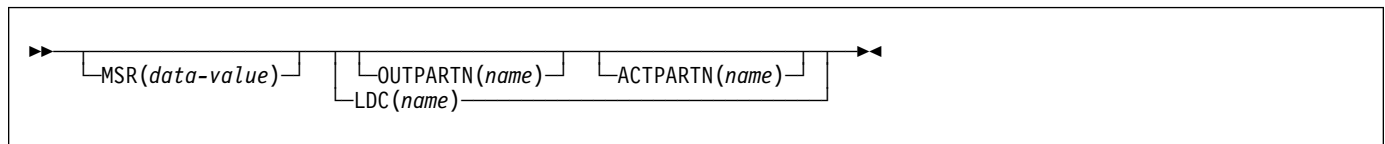
Send device controls to a terminal without map or text data. The keywords are separated into those supported by minimum, standard, and full BMS. For further information about BMS, see the *CICS Application Programming Guide*.

▶▶—SEND CONTROL—◀◀

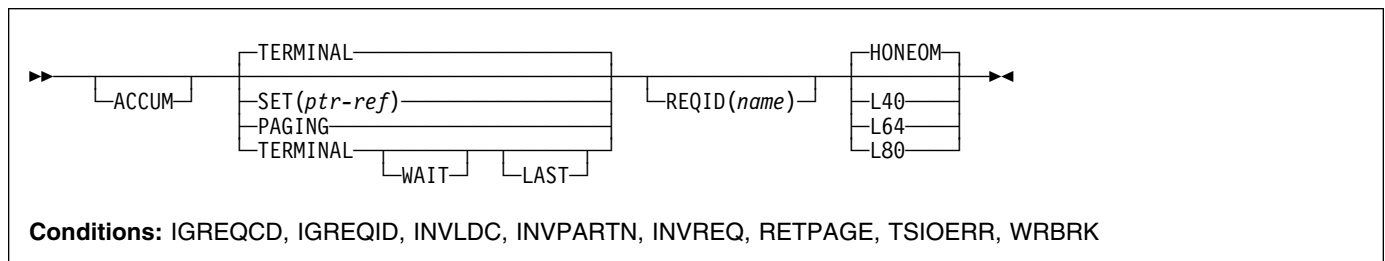
Minimum BMS:



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 428 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.

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.

SEND CONTROL

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 was not used).

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 428 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 DFHMSTRCA to assist in setting this 4-byte area. See "Magnetic slot reader (MSR) control value constants, DFHMSTRCA" on page 413 for a complete list. This option is ignored if the RDO TYPETERM option MSRCONTROL was not used.

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 DFHMDS (see "DFHMDS" on page 433) or the DFHMDSI (see "DFHMDSI" on page 427) 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.

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.

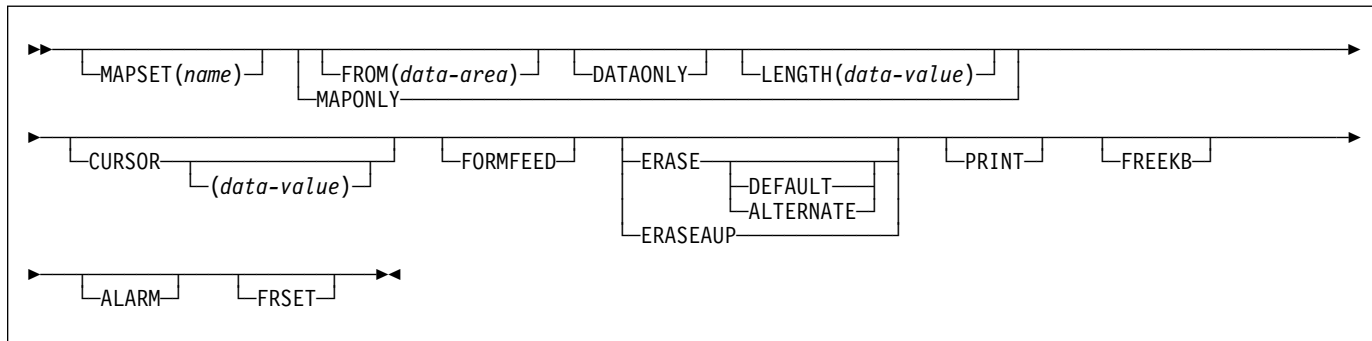
SEND MAP

SEND MAP

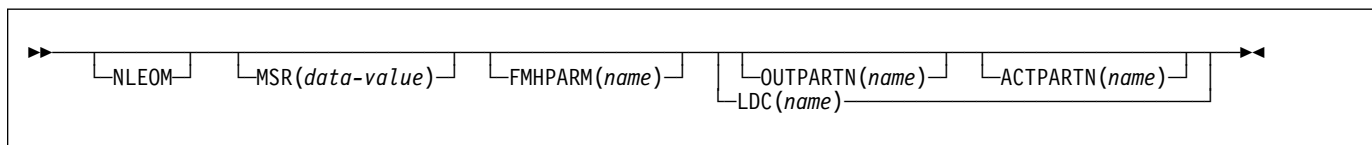
Send mapped output data to a terminal. The keywords are separated into those supported by minimum, standard, and full BMS. For further information about BMS, see the *CICS Application Programming Guide*.

►►SEND MAP(*name*)◄◄

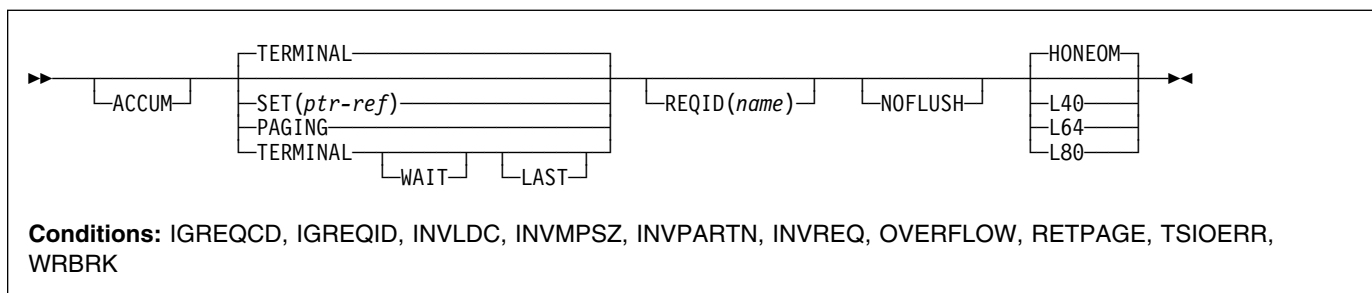
Minimum BMS:



Standard BMS:



Full BMS:



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 428 for a description of the option priority.

See Appendix K, "BMS macro summary" on page 417 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 428 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 was not used).

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 428 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 DFHMDF and DFHMDF BMS map definitions (see pages 431 and 438).

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

SEND MAP

attribute information has been written in the symbolic map.

When using the FRSET option refer to CTRL on page 428 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 428 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 433. 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 431 and 438). 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 428 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 page 413 for a complete list. This option is ignored if the RDO TYPETERM option MSRCONTROL was not used.

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 428 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.

SEND MAP

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

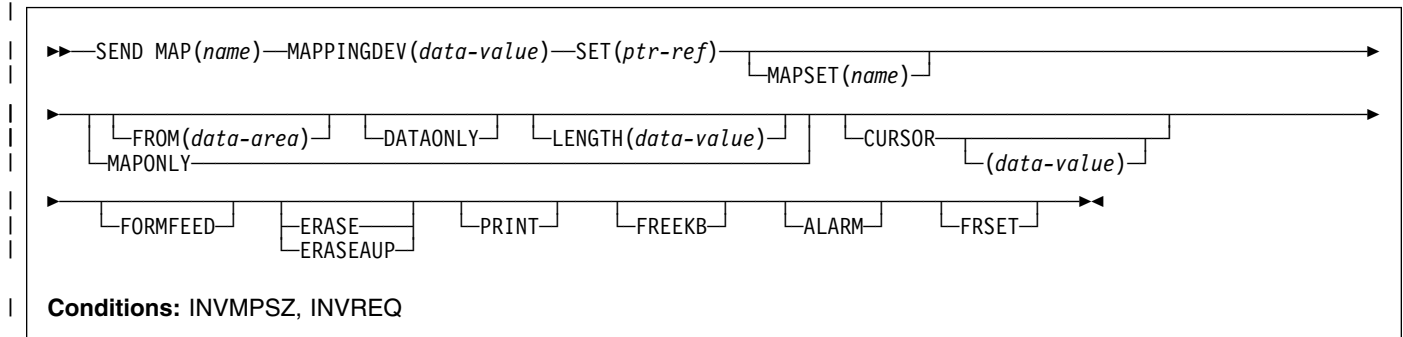
occurs if a SEND MAP command is interrupted by the terminal operator pressing the ATTN key. It applies only to the 2741 Communication Terminal, and only if write break is supported for CICS.

Default action: ignore the condition.

SEND MAP MAPPINGDEV

Create mapped output data to be sent to a terminal described by MAPPINGDEV at some later time. For further information about BMS, see the *CICS Application Programming Guide*.

Minimum BMS:



Description

SEND MAP MAPPINGDEV creates mapped output data to be sent to a terminal that is not the principal facility of the transaction. The terminal characteristics to be used are defined by MAPPINGDEV.

The mapped data is not transmitted but is returned to the application in a buffer defined by the SET option.

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.

When using the ALARM option, refer to CTRL on page 428 for a description of the option priority.

CURSOR(data-value)

specifies the location to which the 3270 cursor is to be returned upon completion of a SEND MAP MAPPINGDEV 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.

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.

ERASE

specifies that the screen printer buffer 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)

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, 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 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 was not used, or the terminal control table TYPE=TERMINAL does not specify FF=YES).

SEND MAP MAPPINGDEV

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 428 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 431 and 438).

FRSET

specifies that the modified data tags (MDTs) of all fields currently in the 3270 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 428 for a description of the option priority.

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 431 and 438). 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.

MAPPINGDEV(*data-value*)

specifies the name of a 3270 terminal whose BMS characteristics match those of the terminal to which the data will eventually be sent using a SEND TEXT MAPPED command or a terminal control SEND or CONVERSE.

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.

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 428 for a description of the option priority.

SET(*ptr-ref*)

specifies the pointer to be set to the address of the mapped data.

The storage area containing the mapped data has the same format as the page buffer returned when using the SET option in the full BMS SEND command. See the description of the MAPPINGDEV facility in the *CICS Application Programming Guide* for more guidance about using the SET option.

Conditions

Some of the following conditions may occur in combination. If more than one occurs, only the first is passed to the application program.

INVMPSZ

occurs if the specified map is too wide for the terminal specified by MAPPINGDEV or if a HANDLE CONDITION OVERFLOW command is active and the specified map is too long for the terminal specified by MAPPINGDEV.

Default action: terminate the task abnormally.

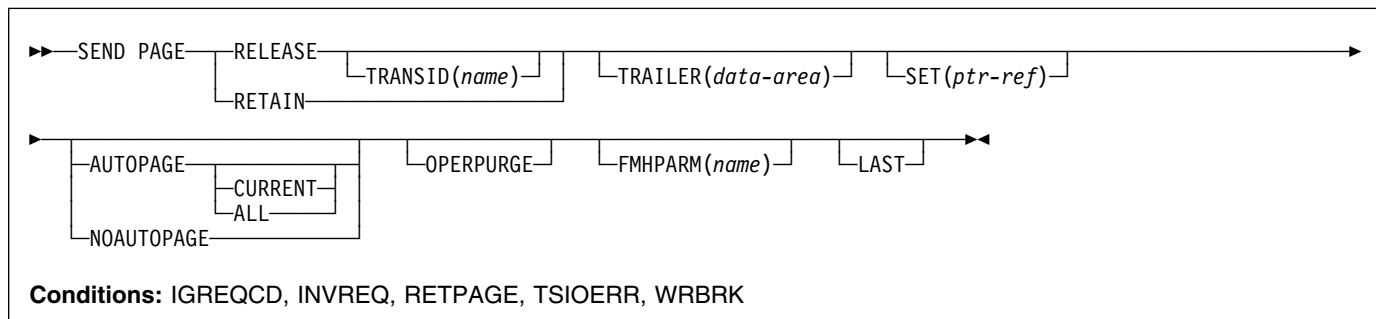
INVREQ

occurs if the terminal specified by MAPPINGDEV does not exist, does not support BMS, or is not a 3270 printer or display.

Default action: terminate the task abnormally.

SEND PAGE

Send last page of data. Only supplied by full BMS. For further information about BMS, see the *CICS Application Programming Guide*.



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.

SEND PAGE

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.

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.

Note: 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.

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.

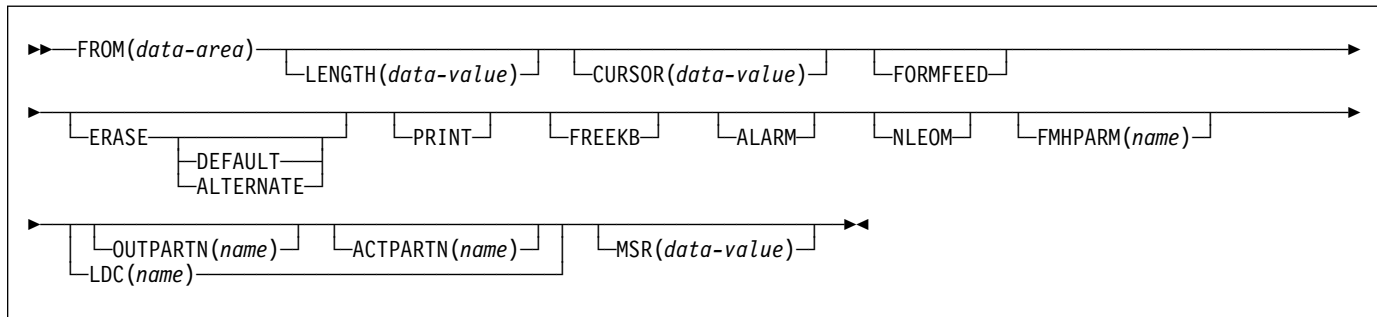
SEND TEXT

SEND TEXT

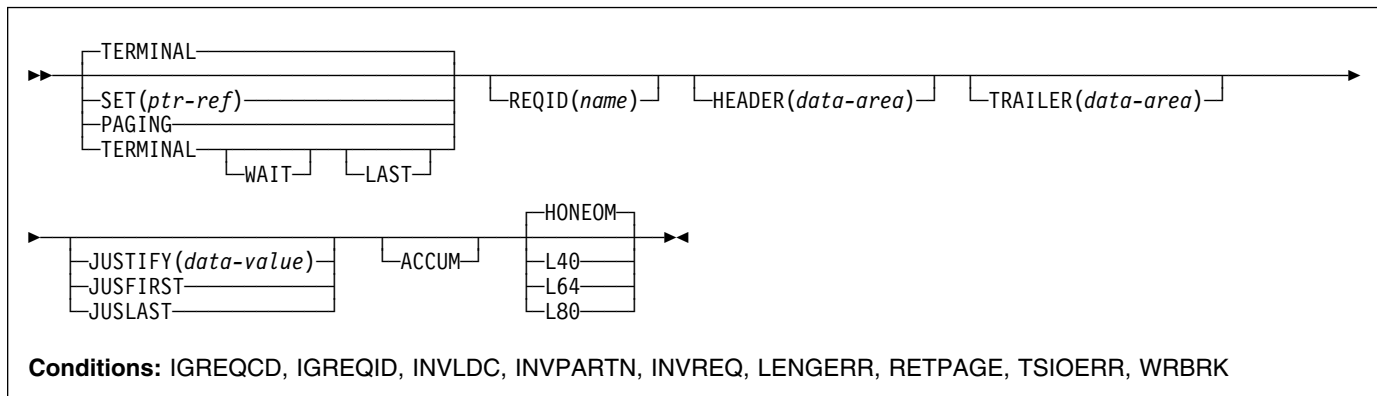
Send data without mapping. The keywords are separated into those supported by standard and full BMS. For further information about BMS, see the *CICS Application Programming Guide*.

▶▶—SEND TEXT—◀◀

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 428 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 was not used).

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 428 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 428 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 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.

SEND TEXT

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 428 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 413 for a complete list. This option is ignored if the RDO TYPETERM option MSRCONTROL was not used.

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 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 428 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

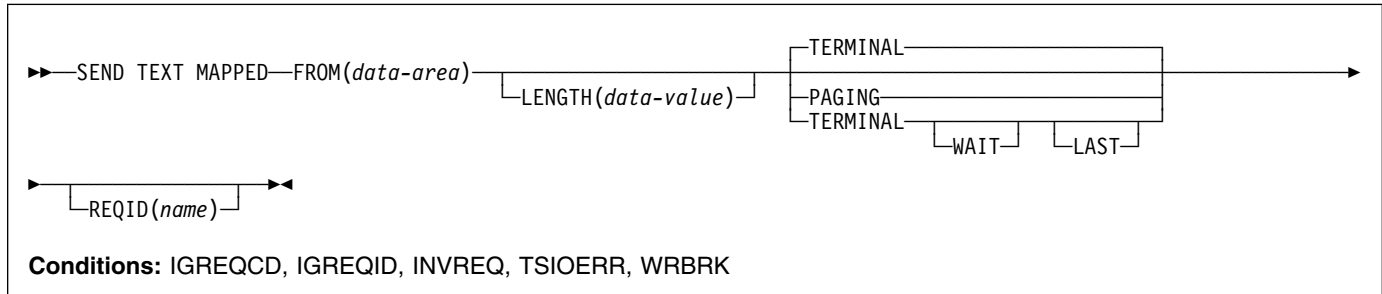
occurs if a SEND command is interrupted by the terminal operator pressing the ATTN key. It applies only to the 2741 Communication Terminal, and only if write break is supported for CICS.

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.

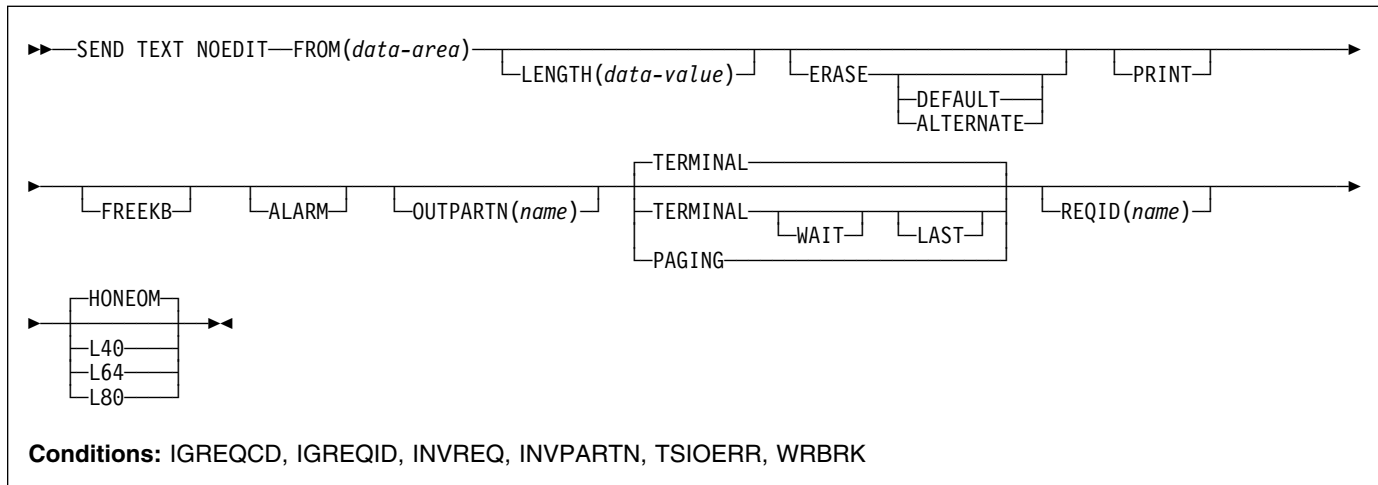
WRBRK

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 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 428 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 428 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 428 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 428 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 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 synchpoint.

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 428 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 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.

SIGNOFF

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, IMS, DB2®, 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

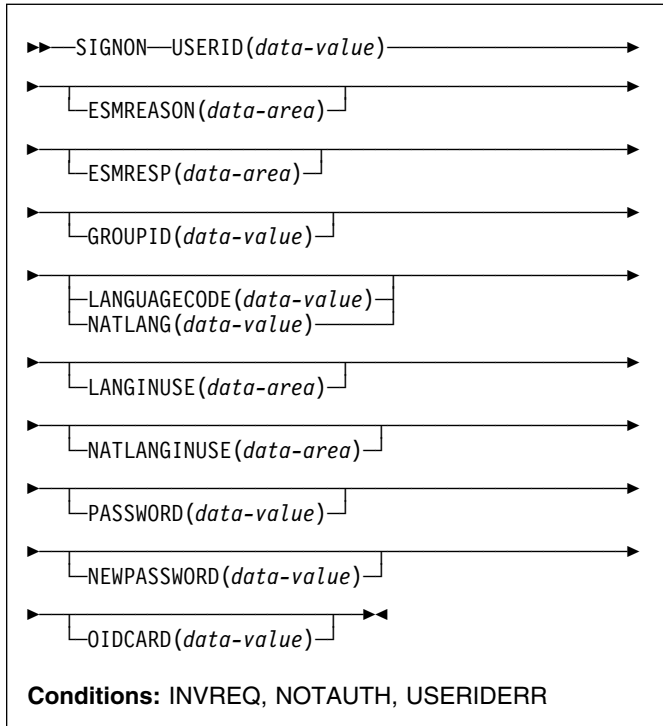
RESP2 values:

- 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

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, IMS, DB2, 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.

If the ESM is RACF, this field is the RACF reason code.

ESMRESP(*data-area*)

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

SIGNON

If the ESM is RACF, this field is the RACF return code.

GROUPID(*data-value*)

assigns, to a RACF 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 RACF database.

If specified, this value, and the USERID, may be propagated to remote systems. See the *CICS RACF Security Guide* for information on implementing LU6.2 security for remote users or MRO Security for remote users..

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 409 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 existing NATLANGINUSE option.

See Appendix I, “National language codes” on page 409 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 409 for possible values of the code.

NATLANGINUSE(*data-area*)

specifies a one character the national language used during the signed-on session. The current implementation always returns the character “E” (U.S. English), which corresponds to the language supplied in the NATLANG option. 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 409 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.
- 26 An error occurred during SNSCOPE checking. The limit of MVS ENQ requests was reached.
- 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 OI DCARD is required.
- 6** The supplied OI DCARD 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 GROU PID is not known to the ESM.
- 24** The USERID is not contained in the GROU PID.

Default action: terminate the task abnormally.

USERIDERR

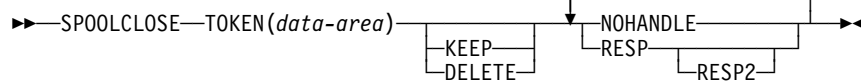
RESP2 values:

- 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 spool report.



Conditions: ALLOCERR, INVREQ, NOSPOOL, NOSTG, NOTFND, NOTOPEN, STRELERR

Description

The SPOOLCLOSE command closes a CICS spool report and, optionally, changes its retention characteristics. If more than one transaction is trying to read reports from JES, SPOOLCLOSE should **not** be immediately followed by SPOOLOPEN. It should be followed by a WAIT, so that other transactions can use the interface.

A default disposition is taken if both KEEP and DELETE are omitted from the SPOOLCLOSE command, or if the report is closed implicitly by a SYNCPOINT or RETURN command:

- When an INPUT report is explicitly closed by a SPOOLCLOSE command, the default disposition is DELETE.
- In all other cases, the default disposition is KEEP.

Options

DELETE

For an INPUT report, DELETE specifies that the **next** report is to be read on the subsequent OPEN INPUT.

For an OUTPUT report, DELETE specifies that the report is to be purged.

Note: When a JCL job is submitted using the internal reader (INTRDR) with the DELETE option specified, the job is sometimes run before the output is deleted.

KEEP

For an INPUT report, KEEP specifies that the report is to be read again when SPOOLOPEN INPUT is next issued.

For an OUTPUT report, KEEP specifies that the report is to be sent to its destination node.

TOKEN(*data-area*)

specifies the 8-character CICS-allocated token used to identify a report.

Conditions

Note: There are no default actions.

ALLOCERR

occurs in any of the following situations:

- Dynamic allocation has rejected a request to allocate an input data set.

RESP2 gives the dynamic allocation response code that denotes this error. The first two characters are the information reason code (S99INFO), and the second two are the error reason code (S99ERROR), as defined in the *MVS/ESA Application Development Guide: Authorized Assembler Language Programs, GC28-1645*.

INVREQ

RESP2 values:

- 4** Unsupported language.
- 8** Unsupported function.
- 40** Subsystem interface already enabled.

Note: Errors 1024 and over are internal, and should not occur. If one of these error codes is returned, contact your IBM support center.

NOSPOOL

RESP2 values:

- 4** No subsystem present.
- 8** Interface being disabled; CICS is quiescing.
- 12** Interface has been stopped.

NOSTG

occurs in any of the following situations:

- A GETMAIN has failed within the JES interface subtask (DFHPSRSS).

RESP2 gives the GETMAIN register 15 return code.

NOTFND

RESP2 values:

1024 Input or output function has been corrupted, and SPOOLCLOSE could not complete.

NOTOPEN

RESP2 values:

8 Data set has not been opened.

STRELERR

occurs in any of the following situations:

- A FREEMAIN has failed within the JES interface subtask (DFHPSPSS).

RESP2 gives the FREEMAIN register 15 return code.

SPOOLOPEN INPUT

Open a spool report.

```

SPOOLOPEN INPUT—TOKEN(data-area)—USERID(data-value)—
└── CLASS(data-value) ─┘
└── NOHANDLE ─┘
└── RESP ─┘
└── RESP2 ─┘

```

Conditions: ALLOCERR, ILLOGIC, INVREQ, NOSPOOL, NOSTG, NOTAUTH, NOTFND, NOTOPEN, OPENERR, SPOLBUSY, SPOLERR, STRELERR

Description

The SPOOLOPEN INPUT command opens a spool report for input from the system spooler to CICS.

It prepares to get (read) an existing spool data set directly using external writer name (USERID) and specified class.

Another task could have allocated a spool file for input. In this case, you should retry after a suitable time interval.

When this command has been successfully executed, you should read the report and proceed to CLOSE as soon as possible, in order to permit other users to use the JES single thread. If SPOOLCLOSE is not issued before transaction end or SYNCPOINT, CICS performs an implicit SPOOLCLOSE KEEP, and writes a message to CSMT to alert the system programmer to the possible unnecessary retention of resources. You should not SPOOLOPEN a data set using this command until you are prepared to process it completely.

This command, if successful, returns a token, which is used later to identify the report in SPOOLREAD and SPOOLCLOSE commands.

Options

CLASS(data-value)

specifies a 1-character class designation. The CLASS operand can be used as a selection parameter for input reports. If it is omitted, the first report for the specified USERID is obtained, regardless of its class.

TOKEN(data-area)

specifies the 8-character CICS-allocated token used to identify a report.

USERID(data-value)

specifies the 8-character user identifier. It must begin with the same 4 characters as the CICS generic applid, so that CICS can check that users are not attempting to access data sets not intended for their CICS system.

Conditions

Note: There are no default actions.

ALLOCERR

occurs in any of the following situations:

- Dynamic allocation has rejected a request to allocate an input data set.

RESP2 gives the dynamic allocation response code that denotes this error. The first two characters are the information reason code (S99INFO), and the second two are the error reason code (S99ERROR), as defined in the *MVS/ESA Application Development Guide: Authorized Assembler Language Programs, GC28-1645*.

ILLOGIC

RESP2 values:

- 3 Invalid CLASS value specified.

INVREQ

RESP2 values:

- 4 Unsupported language.
- 8 Unsupported function.
- 16 USERID missing.
- 36 INPUTIOUTPUT missing.
- 40 Subsystem interface already enabled.

Note: Errors 1024 and over are internal, and should not occur. If one of these error codes is returned, contact your IBM Support Center.

NOSPOOL

RESP2 values:

- 4 No subsystem present.
- 8 Interface being disabled; CICS is quiescing.
- 12 Interface has been stopped.

NOSTG

occurs in any of the following situations:

- A GETMAIN has failed within the JES interface subtask (DFHPSPSS).
RESP2 gives the GETMAIN register 15 return code.

NOTAUTH

occurs in any of the following situations:

- An application has issued a SPOOLOPEN INPUT command with an unauthorized USERID. For the USERID to be authorized, its first four characters must match the first four characters of the current CICS applid id.

NOTFND

RESP2 values:

- 4** No data sets could be located for retrieval for the specified external writer name

| Can also be returned if the CICS region USERID
| does not have ALTER access to the appropriate
| PROFILE in the JESSPOOL class. See the *CICS*
| *RACF Security Guide*. for more information about
| RACF authorization of JES.

- 1024** Input or output function has been corrupted, and SPOOLCLOSE could not complete.

NOTOPEN

RESP2 values:

- 8** Data set has not been opened or a task which has not issued the SPOOLOPEN for a spool data set has attempted to access it.

- 1024** Subtask OPEN macro failure.

OPENERR

RESP2 values:

- 4** A VSAM SHOWCB macro failed to return the lengths of the VSAM control blocks used to access the JES spool file.

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

- An internal error occurred during SPOOLOPEN processing that has forced the request to fail.

SPOLBUSY

RESP2 values:

- 4** Interface already in use by another task.
8 Interface already in use by current task.

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

- The JES/input single thread within the JES interface was not available.

SPOLERR

occurs in any of the following situations:

- The subsystem interface macro (IEFSSREQ) has failed. No input data set name was selected.
RESP2 gives the 'IEFSSREQ' response code.

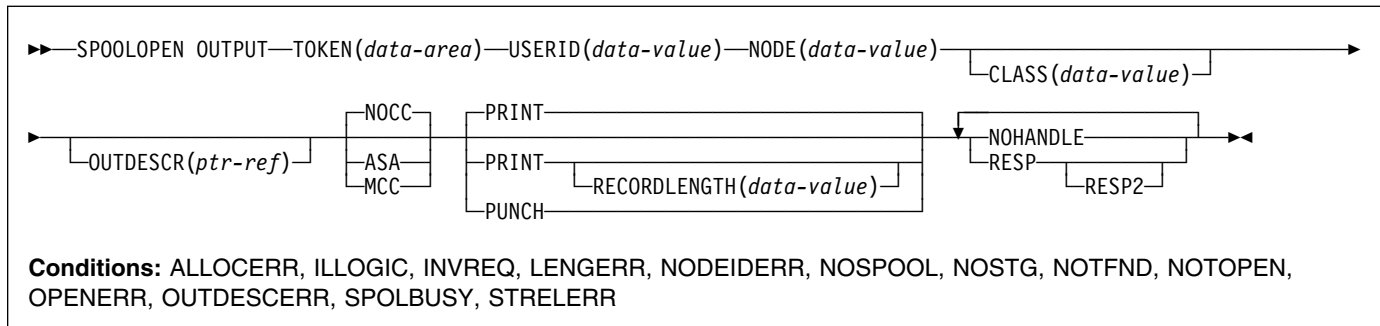
STRELERR

occurs in any of the following situations:

- A FREEMAIN has failed within the JES interface subtask (DFHPSPSS).
RESP2 gives the FREEMAIN register 15 return code.

SPOOLOPEN OUTPUT

Open a spool report for output.



Description

The SPOOLOPEN OUTPUT command opens a spool report for output from CICS to the system spooler and defines its characteristics.

It results in a dynamic allocation of the output file using the nodeid to specify the remote destination and the userid to specify the remote user. As this is a multithread output request, requesters of this service could interleave. This SPOOLOPEN OUTPUT command 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 options to control output formatting. If you do not specify a format, the default value of NOCC is used. NODE and USERID can be used to write the data set directly to the local spool file only if specified with a value of '*'. *

If you do not issue SPOOLCLOSE before the end of the transaction, CICS performs an implicit SPOOLCLOSE DELETE and writes a message to CSMT to alert you to the possible unnecessary retention of resources.

Note: If you retrieve a formatted data set, the system spooler could have changed the data set format. For example, the system spooler could have converted an MCC format data set to ASA format during data set creation. This does not affect the final printed output.

Options

ASA

specifies that the report has each record prefixed with an ASA carriage-control character, and this character must be used by the operating system to control formatting when the report is printed.

CLASS(data-value)

specifies a 1-character class designation. If it is omitted, class A is assumed.

MCC

specifies that the report has each record prefixed with an IBM machine command code carriage-control character, and this character must be used by the operating system to control formatting when the report is printed.

NOCC

specifies that the report has no internal formatting controls. When the report is printed, the operating system prefixes each record with a carriage-control character that causes page skipping according to the default operating system lines-per-page value.

NODE(data-value)

specifies the 8-character identifier of a destination node that the system spooler uses to route the file. It is a sender field.

OUTDESCR(ptr-ref)

(MVS/SP—JES2 Version 3, or JES3 Version 4.2.1 only, or a later upward-compatible release) specifies a pointer variable to be set to the address of a field that contains the address of a string of parameters to the OUTPUT statement of JCL. This is called double indirect addressing. The user must set up the pointer, the address field, and the string. This means that the OUTDESCR option cannot be used from within CECI. The format of the string is:

```
Offset Length Contents
0 4 Length (n) of following text string
4 n OUTPUT statement parameters
```

The parameters use the same keywords and values as the OUTPUT statement but the syntax varies slightly. The following is the format of the OUTDESCR parameter string:

```
keyword1(value1) [keyword2(value2)]
[keyword3(value3,value4)] ...
```

This corresponds to the following OUTPUT statement parameter string:

```
keyword1=value1 [keyword2=value2]
[keyword3=(value3,value4)] ...
```

For details of valid keywords and values, see the *TSO/E Command Reference* manual (SC28-1991-0).

The OUTDESCR operand:

- Can override the NODE and USERID operands only if they are specified with a value of '*'.
- Cannot override the CLASS operand, even if it is omitted and defaults to class A.

Use this operand to set additional attributes for the spool data set.

PRINT

I allows large records (maximum 32760 bytes) to be written to the spool. This is the default setting. This is included for compatibility with the spool support provided with CICS/DOS/VS and CICS Transaction Server for OS/390.

PUNCH

must be specified if the CLASS parameter for the output data set implies punch, and the data set is destined for a VM/RSCS node. This ensures that the record length indicator is set to 80, which is a requirement of VM/RSCS for punch files.

RECORDLENGTH(*data-value*)

specifies, as a halfword binary variable, the maximum length of record to write to a print data set. The default value is 32760.

TOKEN(*data-area*)

specifies the 8-character CICS-allocated token used to identify a report.

USERID(*data-value*)

specifies the 8-character identifier of the destination userid that processes the report. The report carries this identifier, which is used to select the report at its destination. It is a sender field and must be declared with a length of 8 characters.

I Validity checking is performed for USERID. Checks are made for blanks (X'40'), and nulls (X'00').

Sending the internal reader buffer directly to JES:

Instead of waiting for the buffer in your address space to fill up, send the contents of the internal reader buffer directly to JES by coding as your last record:

```
/*E0F
```

This control statement delimits the job in the data set, and makes it eligible for immediate processing.

For more information about using the internal reader, and about other /* control statements, see the *MVS/ESA JCL User's Guide*.

Conditions

Note: There are no default actions.

ALLOCERR

occurs in any of the following situations:

- Dynamic allocation has rejected a request to allocate an input data set.

RESP2 gives the dynamic allocation response code that denotes this error. The first two characters are the information reason code (S99INFO), and the second two are the error reason code (S99ERROR), as defined in the *MVS/ESA Application Development Guide: Authorized Assembler Language Programs, GC28-1645*.

ILLOGIC

occurs in any of the following situations:

- Invalid CLASS value specified.

INVREQ

RESP2 values:

- 4 Unsupported language.
- 8 Unsupported function.
- 16 USERID missing.
- 20 NODE missing.
- 36 INPUT/OUTPUT missing.
- 40 Subsystem interface already enabled.

Note: Errors 1024 and over are internal, and should not occur. If one of these error codes is returned, contact your IBM support center.

- 44 Error in the OUTDESCR string.
- 48 OUTDESCR specified but function not available (wrong level of CICS or JES).
- 52 OUTDESCR specified but bad pointer found on keyword or in OUTDESCR condition.

LENGERR

occurs in any of the following situations:

- RECORDLENGTH not in the range 0 through 32760. RESP2 shows the incorrect value.

NODEIDERR

occurs in any of the following situations:

- JES cannot identify the NODE/USERID combination specified on SPOOLOPEN OUTPUT.

RESP2 gives the dynamic allocation response code that denotes this error. The first two characters are the information reason code (S99INFO), and the second two are the error reason code (S99ERROR), as defined in the *MVS/ESA Application Development Guide: Authorized Assembler Language Programs, GC28-1645*.

SPOOLOPEN OUTPUT

NOSPOOL

RESP2 values:

- 4 No subsystem present.
- 8 Interface being disabled; CICS is quiescing.
- 12 Interface has been stopped.

NOSTG

occurs in any of the following situations:

- A GETMAIN has failed within the JES interface subtask(DFHPSPSS).
RESP2 gives the GETMAIN register 15 return code.

NOTFND

RESP2 values:

- 4 No data sets could be located for retrieval for the specified external writer name.

NOTOPEN

RESP2 values:

- 8 Data set has not been opened.
- 1024 Subtask OPEN macro failure.

OPENERR

RESP2 values:

- 4 A VSAM SHOWCB macro failed to return the lengths of the VSAM control blocks used to access the JES spool file.

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

- An internal error occurred during SPOOLOPEN processing that has forced the request to fail.

OUTDESCRERR

occurs in any of the following situations:

- The macro OUTADD or OUTDEL (invoked as a result of the OUTDESCR specification) failed.
RESP2 gives the reason code from the OUTADD or OUTDEL macro. See the *MVS/ESA Application Development Guide: Authorized Assembler Language Programs, GC28-1645*, for descriptions of codes.

SPOLBUSY

RESP2 values:

- 4 Interface already in use by another task.
- 8 Interface already in use by current task.

Also occurs in any of the following situations:

- The JES/input single thread within the JES interface was not available.

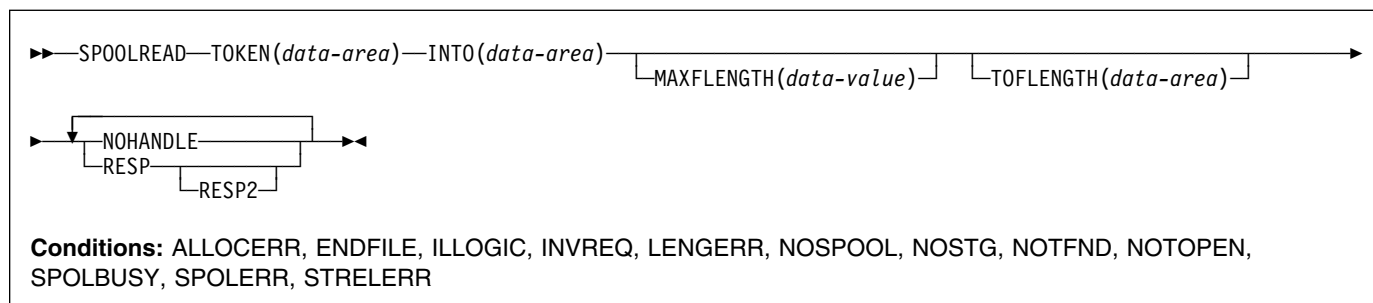
STRELERR

occurs in any of the following situations:

- A FREEMAIN has failed within the JES interface subtask (DFHPSPSS).
RESP2 gives the FREEMAIN register 15 return code.

SPOOLREAD

The SPOOLREAD command obtains the next record from the system spooler.



Options

INTO(*data-area*)

specifies the data area for the variable-length data. It is a receiver field.

MAXLENGTH(*data-value*)

specifies, as a fullword binary variable, the maximum length of data transferred. This is set by the user on input. The limit of **length** is 32 760 bytes. This is the maximum size of the CICS buffer used to read a record.

TOLENGTH(*data-area*)

specifies, as a fullword binary variable, the length of the data that is transferred. This is set by CICS on input. It is optional and, if it is omitted, you are not notified of the actual length of the data received.

TOKEN(*data-area*)

specifies the 8-character CICS-allocated token used to identify a report.

Conditions

Note: There are no default actions.

ALLOCERR

occurs in any of the following situations:

- Dynamic allocation has rejected a request to allocate an input data set.

RESP2 gives the dynamic allocation response code that denotes this error. The first two characters are the information reason code (S99INFO), and the second two are the error reason code (S99ERROR), as defined in the *MVS/ESA Application Development Guide: Authorized Assembler Language Programs, GC28-1645*.

ENDFILE

occurs in any of the following situations:

- All data for the current spool file being read has

been retrieved. You should proceed to issue a SPOOLCLOSE command as soon as possible, to release the lock on the JES single thread, and to terminate current SYSOUT data set processing.

ILLOGIC

RESP2 values:

- 3 Invalid CLASS value specified.

INVREQ

RESP2 values:

- 4 Unsupported language.
- 8 Unsupported function.
- 12 Read attempt after end of file.
- 24 INTO missing.
- 40 Subsystem interface already enabled.

Note: Errors 1024 and over are internal, and should not occur. If one of these error codes is returned, contact your IBM support center.

LENGERR

occurs in any of the following situations:

- You provided insufficient buffer space to read your record, or you requested more than the maximum allowable buffer size of 32 760 bytes (the size of a CICS buffer used to read a record). If the buffer space is too small, it receives as much data as possible. The amount of data truncated is then placed in the RESP2 field. If the TOLENGTH operand is specified, the actual length of the record is placed here.

Note: In the event of a length error due to insufficient buffer space, the next record is not read until the error has been corrected and the current record reread.

RESP2 indicates the amount of data truncated, or shows zero if the MAXLENGTH field is greater

SPOOLREAD

than the maximum allowable buffer size 32 760 bytes.

NOSPOOL

RESP2 values:

- 4 No subsystem present.
- 8 Interface being disabled; CICS is quiescing.
- 12 Interface has been stopped.

NOSTG

occurs in any of the following situations:

- A GETMAIN has failed within the JES interface subtask (DFHPSPSS).
RESP2 gives the GETMAIN register 15 return code.

NOTFND

RESP2 values:

- 4 No data sets could be located for retrieval for the specified external writer name.

NOTOPEN

RESP2 values:

- 8 Data set has not been opened.
- 12 Attempt to read an output file.
- 1024 Subtask OPEN macro failure.

SPOLBUSY

RESP2 values:

- 4 Interface already in use by another task.
- 8 Interface already in use by current task.

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

- The JES/input single thread within the JES interface was not available.

SPOLERR

occurs in any of the following situations:

- The subsystem interface macro (IEFSSREQ) has failed. No input data set name was selected.
RESP2 gives the 'IEFSSREQ' response code.

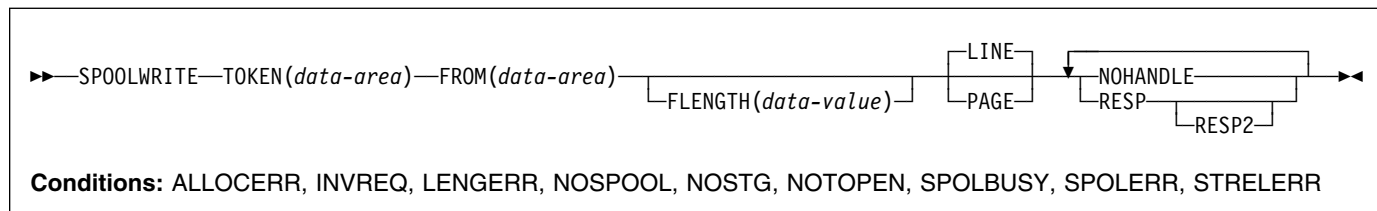
STRELERR

occurs in any of the following situations:

- A FREEMAIN has failed within the JES interface subtask (DFHPSPSS).
RESP2 gives the FREEMAIN register 15 return code.

SPOOLWRITE

The SPOOLWRITE command writes data to a spool report.



Options

FLENGTH(*data-value*)

specifies the fullword binary variable that is to be set to the length of the data that is transferred. This is set by the user on output. It is optional and, if it is omitted, CICS uses the length of the data area.

For OS/VS COBOL users, FLENGTH is mandatory.

FROM(*data-area*)

specifies the data area from which to take the variable length data. The data itself is not altered in any way by CICS. FROM is a sender field.

LINE/PAGE

specifies the format of the data to be sent. The default action is LINE.

The PAGE option must be used to correctly format information for the advanced function printer (AFP™) page printing devices. If a customer is creating MIXED mode type data, that is LINE records and X'5A' (AFPDS or MODCA) pagemode records, the LINE or PAGE operand must match the type record being written to spool.

TOKEN(*data-area*)

specifies the 8-character CICS-allocated token used to identify a report. It is a receiver on SPOOLOPEN and a sender on all other commands.

Conditions

Note: There are no default actions.

ALLOCERR

occurs in any of the following situations:

- Dynamic allocation has rejected a request to allocate an input data set.

RESP2 gives the dynamic allocation response code that denotes this error. The first two characters are the information reason code (S99INFO), and the second two are the error reason code (S99ERROR), as defined in the *MVS/ESA Application*

Development Guide: Authorized Assembler Language Programs, GC28-1645.

INVREQ

RESP2 values:

- 4 Unsupported language.
- 8 Unsupported function.
- 28 FROM missing.
- 40 Subsystem interface already enabled.

Note: Errors 1024 and over are internal, and should not occur. If one of these error codes is returned, contact your IBM support center.

LENGERR

occurs in any of the following situations:

- The value specified in the FLENGTH parameter on a SPOOLWRITE command is not in the valid range 1 to RECORDLENGTH value specified or defaulted at the SPOOLOPEN data set. If the buffer space is too small, it receives as much data as possible.

RESP2 contains the difference between FLENGTH and RECORDLENGTH, or zero if FLENGTH is negative or greater than 32760.

NOSPOOL

RESP2 values:

- 4 No subsystem present.
- 8 Interface being disabled; CICS is quiescing.
- 12 Interface has been stopped.

NOSTG

occurs in any of the following situations:

- A GETMAIN has failed within the JES interface subtask (DFHSPSS).

RESP2 gives the GETMAIN register 15 return code.

NOTOPEN

RESP2 values:

- 8 Spool report has not been opened.

SPOOLWRITE

16 Attempt to write an input file.

1024 Subtask OPEN macro failure.

SPOLBUSY

RESP2 values:

4 Interface already in use by another task.

8 Interface already in use by current task.

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

- The JES/input single thread within the JES interface was not available.

SPOLERR

occurs in any of the following situations:

- The subsystem interface macro (IEFSSREQ) has failed. No input data set name was selected.

RESP2 gives the 'IEFSSREQ' response code.

STRELERR

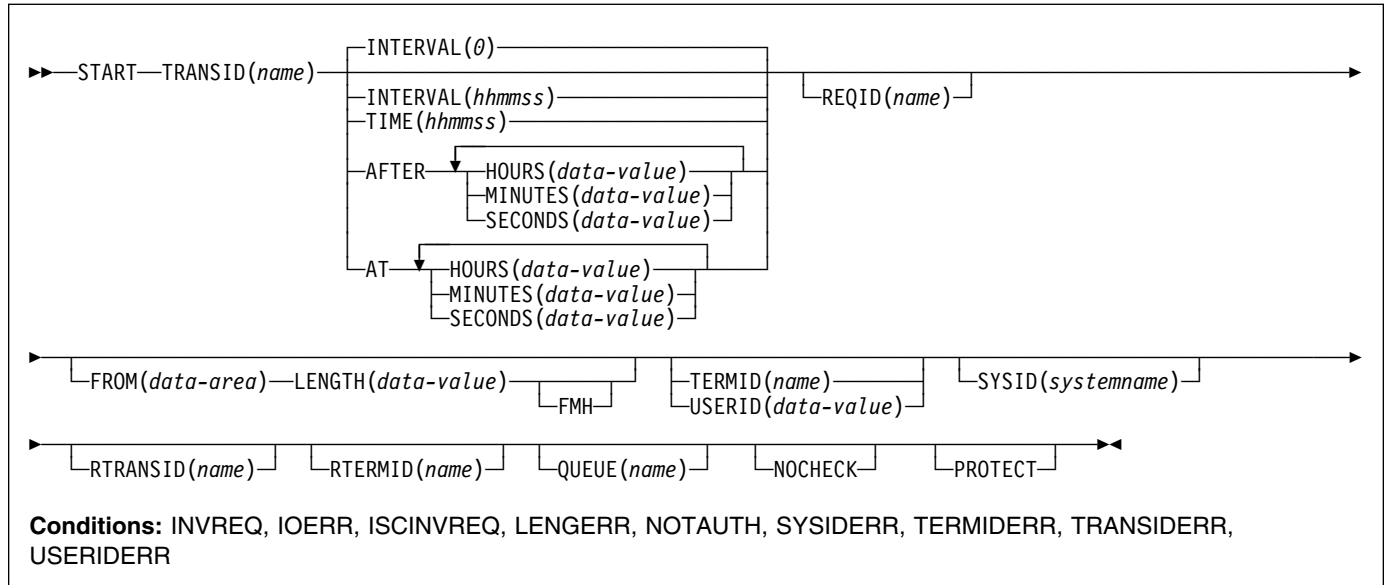
occurs in any of the following situations:

- A FREEMAIN has failed within the JES interface subtask (DFHPSPSS).

RESP2 gives the FREEMAIN register 15 return code.

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, 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

START

options DATASTREAM and RECORDFORMAT. For CICS-CICS, these are always the default values. For CICS-IMS/VSE, care should be taken to specify the correct 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 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 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.

Dynamically routed transactions started by START

commands: Some transactions started by a subset of START commands can be dynamically routed to a remote region. For general information about dynamic transaction routing, and specific information about which transactions started by START commands are eligible for dynamic routing, see the *CICS Intercommunication Guide*.

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 exits) at expiration time.
- 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.

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.

START

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 a name (1–8 characters) that is passed to the
| started task. If this name represents a temporary storage
| queue, the queue must be local to the started task. The
| contents of the queue are not passed.

| 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 RTERMID 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 0–359999 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.

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 **TERMINAL** is not specified). This is referred to as *userid1*.

If you omit both **TERMINAL** 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 HOURS , for AFTER or AT options, is out of range. |
| | 5 | The value specified in MINUTES , for AFTER or AT options, is out of range. |
| | 6 | The value specified in SECONDS , for AFTER or AT options, 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.

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 **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 user exit **XISLCLQ** is enabled (see the *CICS Customization Guide* for programming information).

Default action: terminate the task abnormally.

TERMINERR

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.

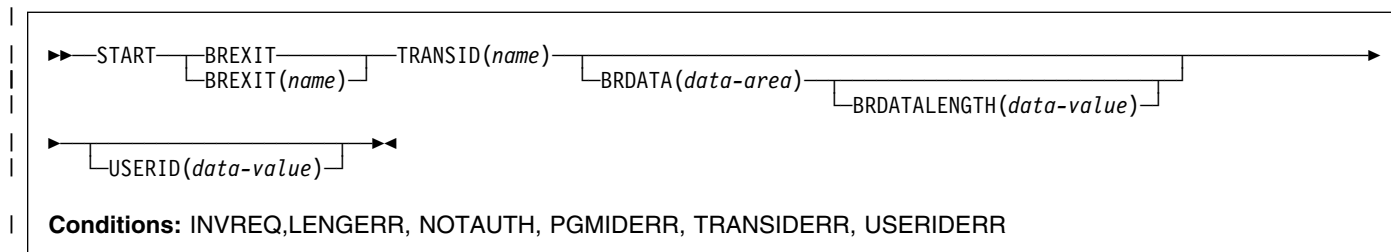
RESP2 values:

11 The specified transaction is defined as remote.

Default action: terminate the task abnormally.

START BREXIT

Start task in the 3270 bridge environment and associate it with the named bridge exit.



Description

START BREXIT starts a task immediately in the local CICS region, and initializes the specified transaction (TRANSID) and bridge exit (BREXIT).

In the 3270 bridge environment, all 3270 terminal requests issued by the transaction specified by TRANSID, are intercepted and passed to the user-replaceable program (the bridge exit) specified by BREXIT.

The bridge exit (BREXIT) emulates the 3270 interface by passing the terminal requests to a client application that may be executing inside or outside of CICS.

See *CICS External Interfaces Guide* for more information about the 3270 bridge and its interfaces.

The attached task cannot be CANCELLED; its STARTCODE is defined by the bridge exit.

Passing data to the bridge exit: Data can be passed to the bridge exit using the BRDATA and BRDATALENGTH options.

The following example shows how to start a specified task, in the 3270 bridge environment and pass data to its bridge exit:

```
EXEC CICS START BREXIT('DFH0CBRE')
          TRANSID('TRNL')
          BRDATA(BRSD)
          BRDATALENGTH(72)
          :
```

Options

BREXIT(name)
specifies the name (1-8 characters) of the bridge exit to be associated with the started task. If no name is specified, the value of BREXIT on the TRANSACTION resource definition for TRANSID is used.

BRDATA(data-area)

specifies the data to be passed to the bridge exit specified by BREXIT when the task is started.

BRDATALENGTH(data-value)

specifies a fullword binary data value that is the length of the BRDATA to be passed to the bridge exit specified by BREXIT when the task is started.

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 BREXIT command. The transaction will be started in the 3270 bridge environment, and will execute in association with the bridge exit specified in BREXIT.

USERID(data-value)

Specifies the userid under whose authority the started transaction is to run.

Conditions

INVREQ

RESP2 values:

11 An attempt was made to route a START BREXIT request.

12 A START BREXIT request has failed..

18 A USERID is specified and the CICS external security manager interface is not initialized.

Default action: terminate the task abnormally.

LENGERR

occurs if BRDATALENGTH is not greater than zero.

Default action: terminate the task abnormally.

NOTAUTH

RESP2 values:

7 A resource security check fails on TRANSID (name).

START BREXIT

| **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.
| Default action: terminate the task abnormally.

| **PGMIDERR**

| occurs if no name is supplied by the BREXIT option and
| the transaction definition for TRANSID does not provide
| a default BREXIT name.
| Default action: terminate the task abnormally.

| **TRANSIDERR**

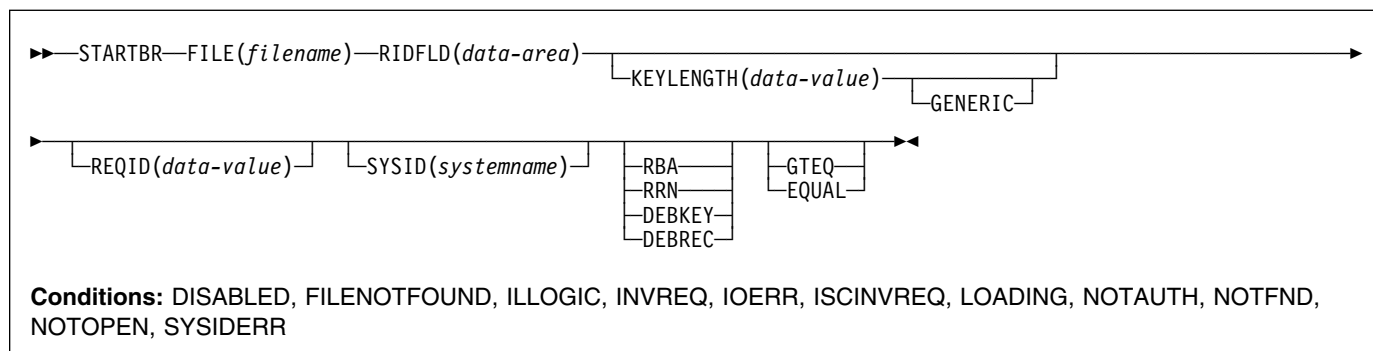
| occurs if the transaction identifier specified in a START
| BREXIT command has not been defined to CICS.
| RESP2 values:
| **11** The specified transaction is defined as remote.
| Default action: terminate the task abnormally.

| **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.

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.

Note: The only data sets greater than 4GB supported by
 # CICS are KSDS, and then only if they are accessed
 # by key. CICS does not support ESDS or RRDS data
 # sets defined with the extended attribute.

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 or RLS 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 BDAM) specifies that deblocking is to occur by key. If neither DEBREC nor DEBKEY is specified, deblocking does not occur.

DEBREC

(blocked BDAM) 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 whether the name is defined in the FCT. Otherwise, the

STARTBR

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.

You cannot use RBA for:

- User-maintained data tables
- Coupling facility data tables
- Any KSDS files opened in RLS access mode
- 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 BDAM 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 377 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 or coupling facility data table and the command does not conform to the format of STARTBR for such a data table (for example, RBA is specified).
- 51 A STARTBR command to a KSDS file that is being accessed in RLS mode specifies the RBA keyword. RLS mode does not support RBA access to KSDS files.

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 377 for details.

For a coupling facility data table, an IOERR indicates a bad response returned from a coupling facility access.

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 The request cannot be satisfied because it is issued against a data table that is still being loaded. The condition can be raised for one of the following reasons:

- The STARTBR specifies a record that has not yet been loaded into a coupling facility data table. Records can be read while a CFDT is loading only if the requested key is within the range of those records already loaded.

The LOADING response can also be returned for a coupling facility data table that has failed during loading. For more information about what happens if the load for a coupling facility data table fails, see the description of the XDTLC global user exit in the *CICS Customization Guide*.

- The READ specifies the GENERIC or GTEQ options for a user-maintained data table. While a UMT is being loaded, you can issue start browse requests with precise keys only.

If your application programs encounter the LOADING condition persistently or too frequently, check that this is not caused by conflicting file definitions that reference the same data set.

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 NOTOPEN (RESP2 60) is returned for one of the following reasons:

- 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. You can also make CLOSED, UNENABLED the initial state, by specifying STATUS(UNENABLED) and OPENTIME(FIRSTREF) on the FILE resource definition. (For BDAM files, you use

SUSPEND

| the FILSTAT parameter of the DFHFCT
| TYPE=FILE macro.)

- | • The requested file is OPEN and in use by
| other transactions, but a CLOSE request
| against the file has been received.
- | • A STARTBR command is issued against a
| data set that is quiesced, or is being
| quiesced, as a result of a SET DSNAME
| QUIESCED or IMMQUIESCED command.

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 CICS region nor a remote system defined by a CONNECTION definition. SYSIDERR also occurs when the link to the remote system is closed.

| **131** For a coupling facility data table, the connection to
| the coupling facility data table server has failed.
| This could be because the server itself has failed,
| or the server is available, but CICS has failed to
| connect to it.

| **132** The start browse is operating on a coupling facility
| data table that no longer exists, probably because
| of a coupling facility failure, in which case the
| coupling facility data table server also fails. See
| the *CICS System Definition Guide* for information
| about restarting a coupling facility data table
| server and reloading a table.

Default action: terminate the task abnormally.

SUSPEND

Suspend a task.

▶▶—SUSPEND—◀◀

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 *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.

UNLOCK

UNLOCK

Release exclusive control.

```
▶—UNLOCK—FILE(filename)—┬—TOKEN(data-area)—┬—SYSID(systemname)—▶
```

Conditions: DISABLED, FILENOTFOUND, ILLOGIC, INVREQ, IOERR, ISCINVREQ, 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 (other than one that refers to a coupling facility data table), 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 the UNLOCK command refers to a record in a recoverable coupling facility data table the record lock is released immediately provided that the task has not made any previous change to the same record (or added it as a new record) within the current unit of work. If changes have been made to the record, or it is a new record added to the table, it remains locked until either a syncpoint command is executed or the task is terminated.

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.

Releasing locks when updating in browse: The UNLOCK command does not release locks held against records locked in response to READNEXT or READPREV commands that specify the update option. It only invalidates the TOKEN value so that it cannot be used to complete an update.

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, READNEXT, or READPREV command that specified the UPDATE option.

If you specify UNLOCK with the TOKEN returned on a READNEXT UPDATE or READPREV UPDATE command for a file accessed in RLS mode, the unlock command simply invalidates the TOKEN value so that it cannot be used to complete an update. It does not release the record lock.

TOKEN can be function shipped. However, if a request specifying TOKEN is function shipped to CICS region that does not support this keyword, the request fails.

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 377 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.

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 377 for details.)

For a coupling facility data table, an IOERR indicates a bad response returned from a coupling facility access.

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** NOTOPEN (RESP2 60) is returned for one of the following reasons:

- 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. You can also make CLOSED, UNENABLED the initial state, by specifying STATUS(UNENABLED) and OPENTIME(FIRSTREF) on the FILE resource definition. (For BDAM files, you use the FILSTAT parameter of the DFHFCT TYPE=FILE macro.)

- The requested file is OPEN and in use by other transactions, but a CLOSE request against the file has been received.
- An UNLOCK command is issued against a data set that is quiesced, or is being quiesced, as a result of a SET DSNAME QUIESCED or IMMQUIESCED command.

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

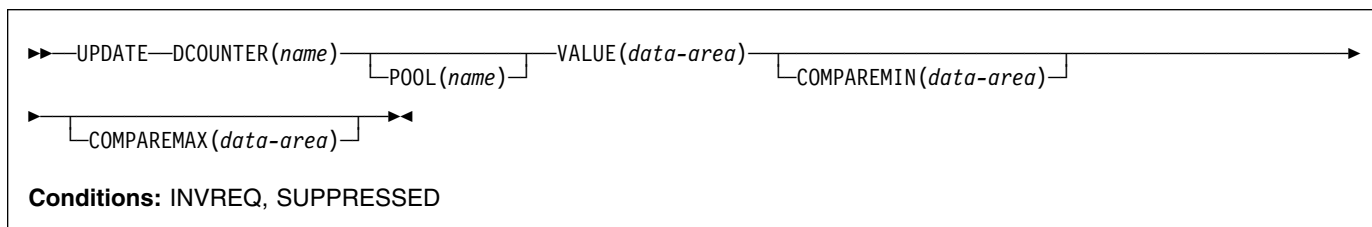
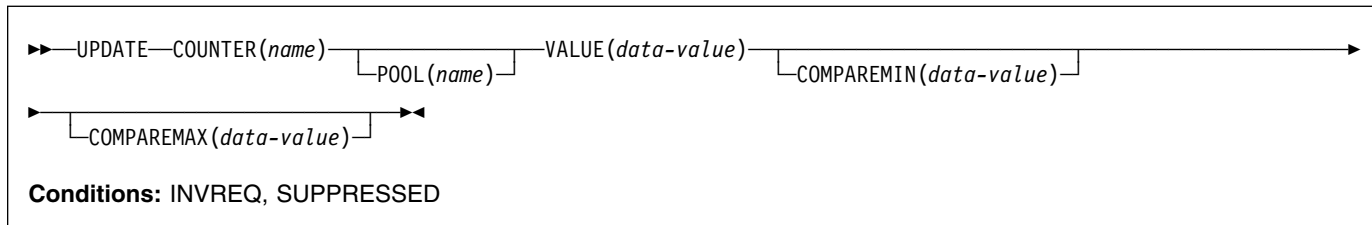
RESP2 values:

- 130** The SYSID option specifies a name that is neither the local CICS region nor a remote system defined to CICS by a CONNECTION definition. SYSIDERR also occurs when the link to the remote system is closed.
- 131** For a coupling facility data table, the connection to the coupling facility data table server has failed. This could be because the server itself has failed, or the server is available, but CICS has failed to connect to it.
- 132** The UNLOCK is issued against a coupling facility data table that no longer exists, probably because of a coupling facility failure, in which case the coupling facility data table server also fails. See the *CICS System Definition Guide* for information about restarting a coupling facility data table server and reloading a table.

Default action: terminate the task abnormally.

UPDATE COUNTER

Update the current value.



Description

These counter commands set a new current value for the named counter. COUNTER operates on fullword signed counters and DCOUNTER on doubleword unsigned counters.

You can use the COMPAREMAX and COMPAREMIN options to set a new current value only if it falls within a specified range, or is above or below a specified value.

For information about specifying fullword and doubleword variables on these named counter commands, see “Argument values” on page 2.

Options

COMPAREMAX(*data-value*)

specifies, as a fullword signed binary value (doubleword unsigned binary value for DCOUNTER), a value to be compared with the named counter's current value, and makes the result of the UPDATE command conditional on the comparison:

- If the current value to be assigned is less than, or equal to, the value specified on the COMPAREMAX parameter, the current value is reset, with response normal
- If the current value is greater than the specified value, CICS returns an exception condition.

The value you specify on the COMPAREMAX parameter can be less than the value on the COMPAREMIN parameter, in which case the current value is considered to be in range if it satisfies the COMPAREMIN or the COMPAREMAX comparison. In the normal case where the COMPAREMIN value is less than the

COMPAREMAX value, the current value must satisfy both comparisons (that is, it must lie between the two values).

COMPAREMIN(*data-value*)

specifies, as a fullword signed binary value (doubleword unsigned binary value for DCOUNTER), a value to be compared with the named counter's current value, and makes the result of the UPDATE command conditional on the comparison:

- If the current value to be assigned is equal to, or greater than, the value specified on the COMPAREMIN parameter, the CICS resets the current value, with response normal
- If the current value is less than the specified value, CICS returns an exception condition.

Note: The value you specify on the COMPAREMIN parameter can be greater than the value on the COMPAREMAX parameter. See the COMPAREMAX parameter for the effect of this.

COUNTER(*name*)

specifies the name of the named counter for which the current number is to be reset to the value specified on the value parameter. The name can be up to 16 alphanumeric characters. If *name* is a variable that contains a name that is less than 16 characters, the name must be padded with trailing blanks.

DCOUNTER(*name*)

specifies the name of the named counter that is to be reset to its minimum value. The name can be up to 16 alphanumeric characters. If *name* is a variable that contains a name that is less than 16 characters, the name must be padded with trailing blanks.

POOL(*poolname*)

specifies the name of the pool in which the named counter resides.

Valid characters for the pool selector string are A through Z, 0 through 9, \$ @ # and _ (underscore). If *name* is a variable that contains a name that is less than 8 characters, the name must be padded with trailing blanks.

This parameter is optional. If you omit the name of the pool, a pool selector value of 8 blanks is assumed.

If there is no matching entry in the DFHNCOPT options table, CICS uses the name specified on the NCPLDFT system initialization parameter, which specifies the default named counter pool.

For information about generating a named counter options table using the DFHNCO macro, see the *CICS System Definition Guide*.

VALUE(*data-value*)

specifies the new number to be set as the current value for the named counter, using a fullword signed binary value for COUNTER and a doubleword unsigned value for DCOUNTER.

Conditions**INVREQ**

RESP2 values:

- 201** Named counter not found.
- 301** The server has reported an error code that is not understood by the named counter interface. Generally, this is not possible unless the interface load module, DFHNCIF, is at a lower maintenance or release level than the server itself.
- 303** An unexpected error, such as structure failure or loss of connectivity, has occurred on a macro used to access the coupling facility. Further information can be found in message DFHNC0441 in the CICS job log.
- 304** The pool selection parameter specified in the program cannot be resolved to a valid server name using the current options table.
- 305** The interface is unable to establish a connection to the server for the selected named counter pool. Further information can be found in an AXM services message (AXMSC*nnnn*) in the CICS job log.
- 306** An abend occurred during server processing of a request. Further information can be found in a message in the CICS job log and the server job log.
- 308** The DFHNCOPT options table module, required for resolving a pool name, could not be loaded.

309 During processing of the options table, the named counter interface encountered an unknown entry format. Either the options table is not correctly generated, or the DFHNCIF interface load module is not at the same release level as the options table.

310 An options table entry matching the given pool name specified a user exit program, but the user exit program is not link-edited with the options table and cannot be loaded.

403 The POOL parameter contains invalid characters or embedded spaces.

404 The COUNTER parameter contains invalid characters or embedded spaces.

406 The VALUE parameter is invalid. You cannot set the current value to less than the minimum value, or greater than the maximum value plus 1.

Default action: terminate the task abnormally.

SUPPRESSED

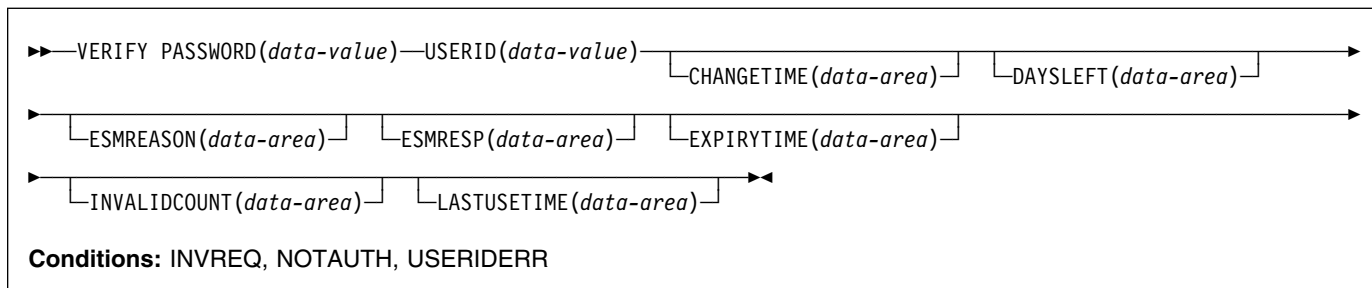
RESP2 values:

- 103** The current value of the named counter is:
- not within the range specified by the COMPAREMAX and COMPAREMIN parameters, when both are specified
 - greater than the COMPAREMAX parameter or less than the COMPAREMIN parameter, when only one option is specified.

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.



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.

When the external security manager is RACF, the CHANGETIME and EXPIRYTIME outputs always show as midnight.

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 in one or more RACF group connections, 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.

When the external security manager is RACF, the time is shown as midnight.

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.

If the ESM is RACF, this field is the RACF reason code.

ESMRESP(data-area)

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

If the external security manager is RACF, this field is the RACF return code.

EXPIRYTIME(data-area)

returns the date and time the password will expire, in ABSTIME units.

When the external security manager is RACF, the time is shown as midnight.

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.

If a user has a never-expiring password that was established with the RACF PASSWORD USER(userid) NOINTERVAL

command, the outputs DAYSLEFT and EXPIRYTIME have little meaning and are shown as -1.

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.

Default action: terminate the task abnormally.

NOTAUTH

RESP2 values:

- 2 The supplied password is wrong. If the external security manager is RACF, the revoke count maintained by RACF is incremented.
- 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:

WAIT CONVID (APPC)

- 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 for an event to occur.

```
▶▶—WAIT EVENT—ECADDR(ptr-value)—┐
                                └─NAME(name)—◀◀
```

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

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 189. 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 ESA WAIT ESA POST ESA WAIT

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

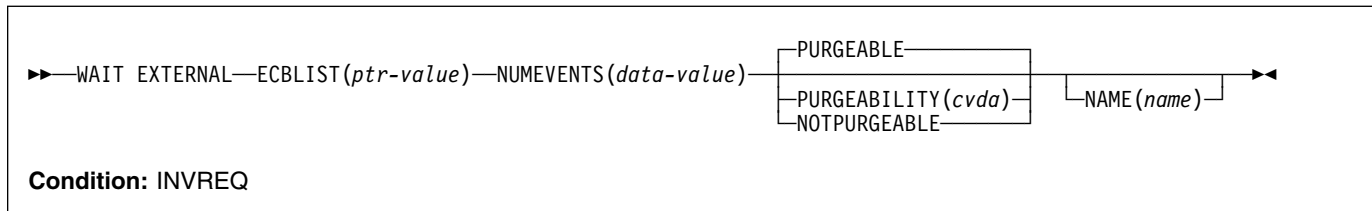
The following example shows you how to suspend the processing of a task until the specified event control area is posted:

```
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 MVS-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 uses extended ECBs and the MVS POST exit mechanism for ECBs passed by WAIT EXTERNAL; therefore do not use WAIT EXTERNAL unless you are sure that the ECBs are not posted by any method other than the MVS POST service or the standard 'optimized post' logic using a compare-and-swap (CS) instruction. Note that the standard 'optimized post' logic is only applicable when the ECB is not waiting, that is when the wait bit X'80' is not on.

If a WAIT EXTERNAL ECB is hand posted, for example by another task moving a value into the ECB, unpredictable errors occur. If there is any possibility of hand posting, use the WAITCICS command. Use WAIT EXTERNAL whenever possible, because it usually has less overhead.

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 MVS-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 INQ TASK command or a CEMT INQ 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:

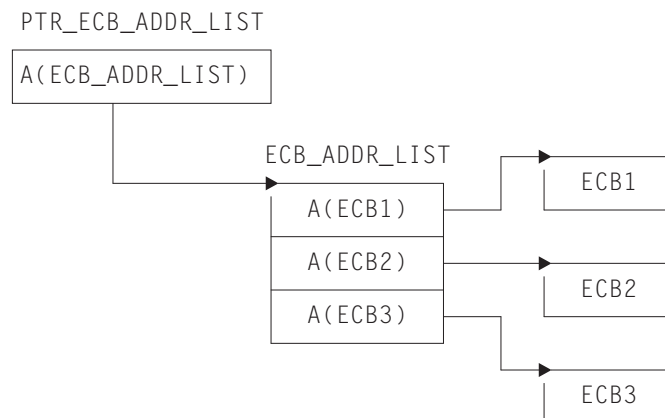
- 1 An ECB is not valid, for example the ECB is not fullword aligned.
- 2 The ECB address is a null pointer, (X'00000000') or (X'FF000000').
- 3 NUMEVENTS is not a positive number.
- 4 PURGEABILITY is specified with an incorrect CVDA.
- 5 No valid ECBs have been found in the list, because either the ECBLIST address is not valid or all the ECB addresses are not valid.

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 5. ECBLIST option, EXEC CICS WAIT EXTERNAL

WAIT JOURNALNAME

Synchronize the task with journal output being written to a system logger log stream.

```
▶▶—WAIT JOURNALNAME(data-value)—┐
                               └─REQID(data-value)—◀◀
```

Conditions: IOERR, JIDERR, NOTOPEN

Description

WAIT JOURNALNAME 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 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 been output to the system logger log stream. This may be done either by issuing a stand-alone WAIT JOURNALNAME command, or by making the last output command itself synchronous (by specifying the WAIT option in the WRITE JOURNALNAME command).

Options

JOURNALNAME(*data-value*)

specifies a 1– to 8– character journal name identifying the journal on which the task is to wait for synchronization. The name must be a journal name known to CICS.

To issue the wait against the CICS system log, specify DFHLOG as the journal name.

To issue the wait against journals defined using the journal numbering convention, as in file resource definitions, specify the name as DFHJ*nn*, where *nn* is the journal number in the range 1 to 99.

Note: Specifying DFHJ01 on this command refers to a user journal, *not* the system log.

REQID(*data-value*)

specifies, in a fullword binary variable, the token that refers to a journal record that has been created but possibly not yet written out. The token is returned by

CICS from a previous WRITE JOURNALNAME command issued by this task.

If you do not specify REQID, the task is synchronized with the output of the current buffer for the journal specified by JOURNALNAME.

Conditions

IOERR

a journal record was not output because of an irrecoverable error condition returned by the system logger or by SMF.

Default action: If the log is the system log, CICS either quiesces or abends. If it is a general log, the task is terminated abnormally.

JIDERR

occurs in either of the following situations:

- The specified journal name is not known in the CICS region.
- The specified journal name refers to a DASD-only log stream to which a CICS region in another MVS image is currently connected.

Default action: terminate the task abnormally.

NOTOPEN

occurs in any of the following situations:

- The command cannot be satisfied because the user explicitly disabled the specified journal.
- A wait request is issued against a journal that has not previously been written to.
- This journal was defined using a model that maps it into the logstream used by the system log for this system. This error is not detected when trying to connect to the logstream. A definition for the journal will be installed and set to failed.

Default action: terminate the task abnormally.

Examples

The following example shows how to request synchronization with the output of journal records written to a user journal named 'ACCOUNTS'

```
EXEC CICS WAIT JOURNALNAME('ACCOUNTS')  
      REQID(RECTOKEN)
```

WAIT JOURNALNUM

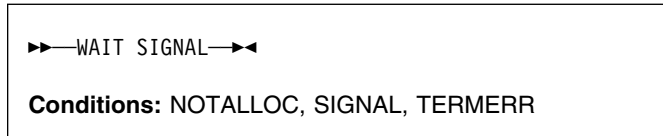
Synchronize with journal output.

Note: This command is supported for compatibility with earlier releases of CICS. It is superseded by the WAIT JOURNALNAME command, which you are recommended to use instead.

The syntax is the same as for WAIT JOURNALNAME except that JOURNALNUM specifies a numeric instead of a character value. The numeric value, nn, is in the range 01–99 and corresponds to journal name **DFHJnn**.

WAIT SIGNAL

Suspend task on a logical unit.



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

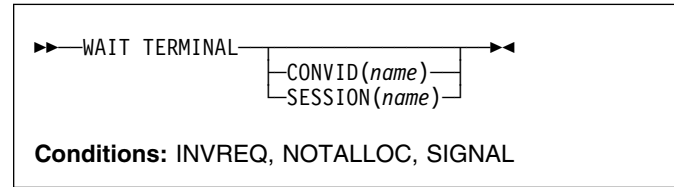
occurs for a terminal-related error.

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 with abend 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.

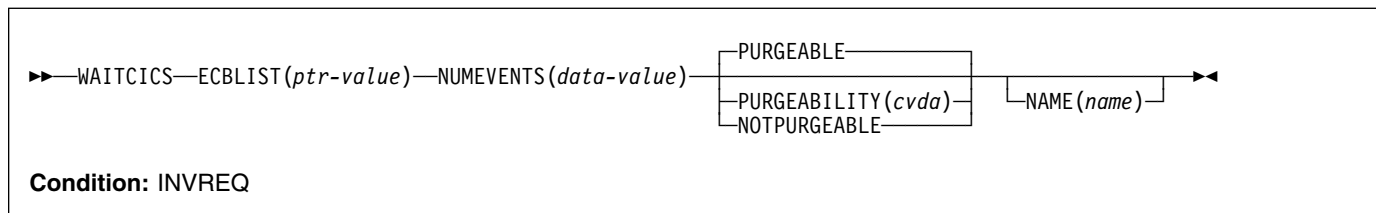
SIGNAL

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 MVS-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 MVS WAIT facility when it runs out of work. Such ECBs can be posted using the MVS 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 MVS WAIT contains duplicate ECB addresses, MVS abends CICS.

Options

ECBLIST(ptr-value)

is a pointer to a list of addresses of MVS-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 INQ TASK or CEMT INQ 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 *CICS System Programming Reference* for information about SET TASK PURGEIFORCEPURGE.

Conditions

WEB ENDBROWSE FORMFIELD

INVREQ

RESP2 values:

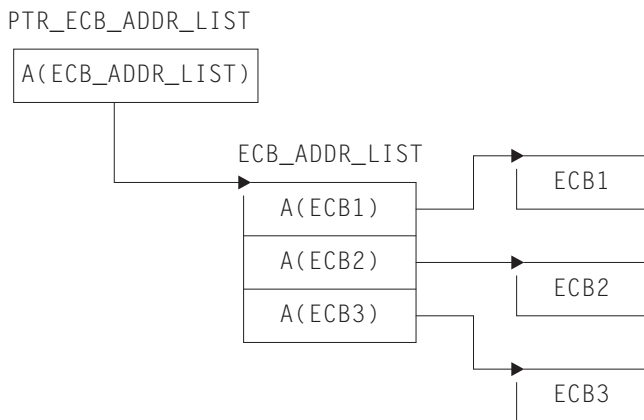
- 1 An ECB is not valid, for example the ECB is not fullword aligned.
- 3 NUMEVENTS is not a positive number.
- 4 PURGEABILITY is specified with an incorrect CVDA.
- 5 No valid ECBs have been found in the list, because either the ECBLIST address is not valid, or all the ECB addresses are not valid.

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
1 ECB_ADDR_LIST,
2 ECB_ADDR(3) PTR;
DCL
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 6. ECBLIST option, EXEC CICS WAITCICS

WEB ENDBROWSE FORMFIELD

Signal end of FORMFIELD browse.

```

#
# ►►—WEB—ENDBROWSE—FORMFIELD—◄◄
#
# Conditions: INVREQ
  
```

Description

WEB ENDBROWSE FORMFIELD terminates the browse of a # set of name-value pairs in an HTML form. The form is part of # the body of an HTTP request being processed by the current # CICS task. No information is returned on the ENDBROWSE.

Conditions

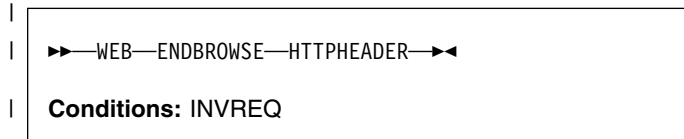
INVREQ

occurs for the following conditions. RESP2 values are:

- # 1 The command is being issued in a non-CICS Web application.
- # 3 The command is being issued for a non-HTTP request.
- # 4 The command is being issued before a WEB STARTBROWSE command is issued.

WEB ENDBROWSE HTTPHEADER

Signal end of HTTP header browse.



Description

WEB ENDBROWSE HTTPHEADER terminates the browse of HTTP headers. No information is returned on the ENDBROWSE.

Conditions

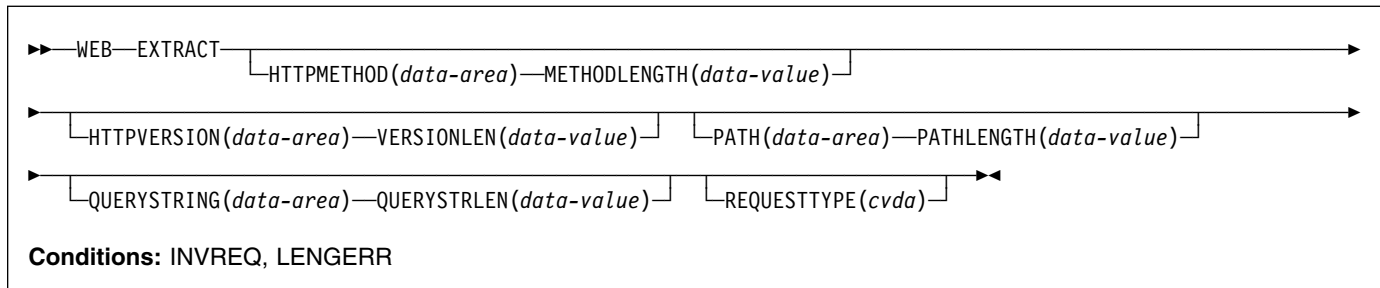
INVREQ

occurs for the following conditions. RESP2 values are:

- 1** The command is being issued in a non-CICS Web interface application.
- 3** The command is being issued for a non-HTTP request.
- 4** The command is being issued before a WEB STARTBROWSE command is issued.

WEB EXTRACT

Obtain information about inbound client request.



Description

WEB EXTRACT allows the application to obtain additional information about the inbound request from the client.

Note: This command can also be specified as EXTRACT WEB.

Options

HTTPMETHOD(data-area)

specifies a buffer to contain the HTTP method string on the request line of the inbound message.

HTTPVERSION(data-area)

specifies a buffer to contain the HTTP version string on the request line of the inbound message.

METHODLENGTH(data-value)

specifies the length of the buffer supplied on the HTTPMETHOD option, and is set to the actual length of the data returned to the application. If the data exceeds the buffer length, a LENGERR condition is raised and the data is truncated. The actual length of the HTTPMETHOD is returned in METHODLENGTH.

PATH(data-area)

specifies a buffer to contain the PATH specified in the request line of the inbound message.

PATHLENGTH(data-value)

specifies the length of the buffer supplied on the PATH option, and is set to the actual length of the data returned to the application. If the data exceeds the buffer length, a LENGERR condition is raised and the data is truncated. The actual length of the PATH is returned in PATHLENGTH.

QUERYSTRING(data-area)

specifies a buffer to contain the query string on the request line of the HTTP request. The query string is the value or values encoded after the question mark (?)

delimiting the end of the path on the request line of an HTTP request. The data is passed to the application in its escaped form.

QUERYSTRLEN(data-value)

specifies the length of the buffer supplied on the QUERYSTRING option. On returning from CICS, QUERYSTRLEN is set to the actual length of the data returned to the application. If the data exceeds the buffer length, a LENGERR condition is returned and the data is truncated.

REQUESTTYPE(cvda)

specifies the type of request received. The request type can be HTTPYES or HTTPNO.

Conditions

INVREQ

occurs for the following conditions. RESP2 values are:

- 1 The command is being issued in a non-CICS Web interface application.
- 3 The command is being issued for a non-HTTP request (this is set only if one or more of HTTPMETHOD, HTTPVERSION, or PATH is specified and the request is a non-HTTP request).

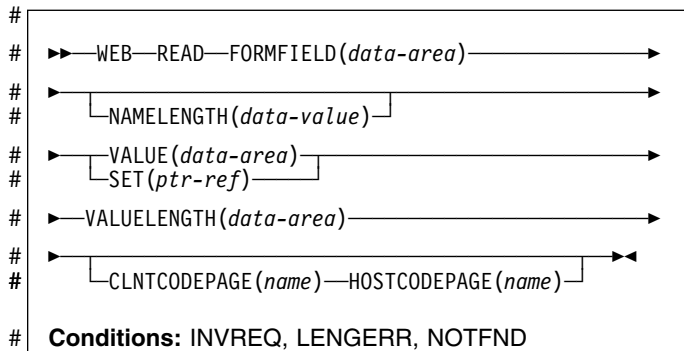
LENGERR

occurs for the following conditions. RESP2 values are:

- 4 The length in METHODLENGTH is less than or equal to zero.
- 5 The length in VERSIONLEN is less than or equal to zero.
- 6 The length in PATHLENGTH is less than or equal to zero.
- 8 The length in QUERYSTRLEN is less than or equal to zero.

WEB READ FORMFIELD

Retrieve the value of a field from an HTML form.

**# Description**

WEB READ FORMFIELD retrieves the value of a specific # field from an HTML form, the name of which is given on the # request in the FORMFIELD parameter. The form is part of # the body of an HTTP request being processed by the current # CICS task.

Options**# CLNTCODEPAGE(name)**

specifies the 40-character name of the codepage to be # used when data is converted from the client codepage. If # this is not specified, CICS obtains it from the charset # parameter on the Content-Type header of the HTTP # request for the current CICS task. If the Content-Type # header is not present, CICS used codepage 819 # (ISO-8859-1). If you specify CLNTCODEPAGE you # must also specify HOSTCODEPAGE. Valid values for # CLNTCODEPAGE are the CICS-supported CLINTCP # and SRVERCP values described in *CICS Family: # Communicating from CICS on System/390*. Mapping is # performed between the IANA Coded Character Set # values used in HTTP requests to the CLINTCP values # known to CICS; these values are listed in the *CICS # Internet Guide*.

FORMFIELD(data-area)

specifies the name of the form field to extract. It is a # string of text containing the name of the requested field. # The string of text supplied is not case sensitive.

HOSTCODEPAGE(name)

specifies the 8-character name of the host codepage to # be used when the forms data is converted from the # ASCII codepage, it was received in, into the EBCDIC, # CICS processes it in. If this is not specified the default

(037) is used. If you specify HOSTCODEPAGE you # must also specify CLNTCODEPAGE.

NAMELENGTH(data-value)

specifies the length, as a fullword binary value, of the # form field name.

SET(ptr-ref)

specifies a pointer reference that is to be set to the # address of data received. The pointer reference is valid # until the end of the task.

VALUE(data-area)

specifies the buffer to contain the value of the named # form field. CICS unescapes any escaped characters # before placing them in the buffer.

VALUELENGTH(data-area)

specifies the length, as a fullword binary value, of the # buffer that is to contain the form field value. If you # specify the VALUE option, VALUELENGTH specifies the # maximum length of the data that the program accepts. If # the value exceeds the length of the buffer, it is # truncated. If the length of the form field value is less # than the size of the buffer, the form field value is placed # in the leftmost byte positions.

Conditions**# INVREQ**

RESP2 values are:

- # 1 The command is being issued in a non-CICS Web application.
- # 3 The command is being issued for a non-HTTP request.
- # 11 The client codepage cannot be found.
- # 12 The server codepage cannot be found.
- # 13 No forms data has been supplied in the body of the HTTP request.
- # 14 The codepage combination for client and server is invalid.

LENGERR

RESP2 values are:

- # 1 The length in VALUELENGTH is less than or equal to zero.
- # 2 The form field value has been truncated during a read operation because the receiving buffer is too small.

NOTFND

RESP2 values are:

- # 1 The form field with the given name cannot be found.

WEB READ HTTPHEADER

Extract HTTP header information.

```
►►WEB—READ—HTTPHEADER(data-area)—NAMELENGTH(data-value)—VALUE(data-area)—VALUELENGTH(data-value)►►
```

Conditions: INVREQ, LENGERR, NOTFND

Description

WEB READ HTTPHEADER extracts HTTP header information.

Options

HTTPHEADER(*data-area*)

specifies the name of the HTTP header.

NAMELENGTH(*data-area*)

specifies the length, as a fullword binary value, of the HTTP header name.

VALUE(*data-area*)

specifies the buffer to contain the value of the HTTP header being extracted.

VALUELENGTH(*data-area*)

specifies the length of the buffer supplied on the VALUE option, and is set to the actual length of the data returned to the application. If the data exceeds the buffer length, a LENGERR condition is raised and the data is truncated. The actual length of the VALUE is returned in VALUELENGTH.

Conditions

INVREQ

RESP2 values are:

- 1 The command is being issued in a non-CICS Web interface application.
- 3 The command is being issued for a non-HTTP request.

LENGERR

RESP2 values are:

- 1 The length in VALUELENGTH is less than or equal to zero.
- 2 The header value has been truncated because the receiving buffer is too small.

NOTFND

RESP2 values are:

- 1 The header with the given name could not be found.

WEB READNEXT FORMFIELD

Retrieve next name-value pair in an HTML form.

```
#
# ►►WEB—READNEXT—FORMFIELD(data-area)—NAMELENGTH(data-area)—VALUE(data-area)—VALUELENGTH(data-area)►►
#
# Conditions: ENDFILE, INVREQ, LENGERR
```

Description

WEB READNEXT FORMFIELD retrieves the next name-value pair in an HTML form.

6 A form field has been found that is not in the expected format.

LENGERR

RESP2 values are:

Options**# FORMFIELD(*data-area*)**

specifies the buffer that contains the form field. The case of the name is the same as it is stored in the form.

1 NAMELENGTH or VALUELENGTH is less than or equal to zero.

NAMELENGTH(*data-area*)

specifies the length, as a fullword binary value, of the form field name. If the length of the form field value is less than the size of the buffer, the form field value is placed in the leftmost byte positions. NAMELENGTH contains the actual length of the data returned.

4 The form field name has been truncated during a browse operation because the receiving buffer is too small.

5 The form field value has been truncated because the receiving buffer is too small.

VALUE(*data-area*)

specifies the buffer that contains the value corresponding to the name returned in the FORMFIELD data area. CICS unescapes any escaped characters before placing them in the buffer.

VALUELENGTH(*data-area*)

specifies the length, as a fullword binary value, of the buffer that contains the form field value. If the value exceeds the buffer length, it is truncated. If the length of the form field value is less than the size of the buffer, the form field value is placed in the leftmost byte positions. VALUELENGTH contains the actual length of the data returned.

Conditions**# ENDFILE**

The end of the list of name/value pairs has been reached.

INVREQ

RESP2 values are:

1 The command is being issued in a non-CICS Web application.

3 The command is being issued for a non-HTTP request.

4 The command is being issued before a WEB STARTBROWSE FORMFIELD has been issued.

WEB READNEXT HTTPHEADER

Retrieve next HTTP header.

►►—WEB—READNEXT—HTTPHEADER(*data-area*)—NAMELENGTH(*data-value*)—VALUE(*data-area*)—VALUELENGTH(*data-value*)—◄◄

Conditions: ENDFILE, INVREQ, LENGERR

Description

WEB READNEXT HTTPHEADER retrieves the next HTTP header in the list of headers.

Options

HTTPHEADER(*data-area*)
 specifies the buffer to contain the name of the HTTP header being extracted.

NAMELENGTH(*data-area*)
 # specifies the length of the buffer supplied on the HTTPHEADER option, and is set to the actual length of the data returned to the application. If the data exceeds the buffer length, a LENGERR condition is raised and the data is truncated. The actual length of the HTTPHEADER name is returned in NAMELENGTH.

VALUE(*data-area*)
 specifies the buffer to contain the value of the HTTP header being extracted.

VALUELENGTH(*data-area*)
 # specifies the length of the buffer supplied on the VALUE option, and is set to the actual length of the data returned to the application. If the data exceeds the buffer length, a LENGERR condition is raised and the data is truncated. The actual length of the VALUE is returned in VALUELENGTH.

Conditions

ENDFILE
 The end of the list of HTTPHEADERS has been reached.

INVREQ
 RESP2 values are:

- 1** The command is being issued in a non-CICS Web interface application.
- 3** The command is being issued for a non-HTTP request.
- 4** The command is being issued before a WEB STARTBROWSE has been issued.

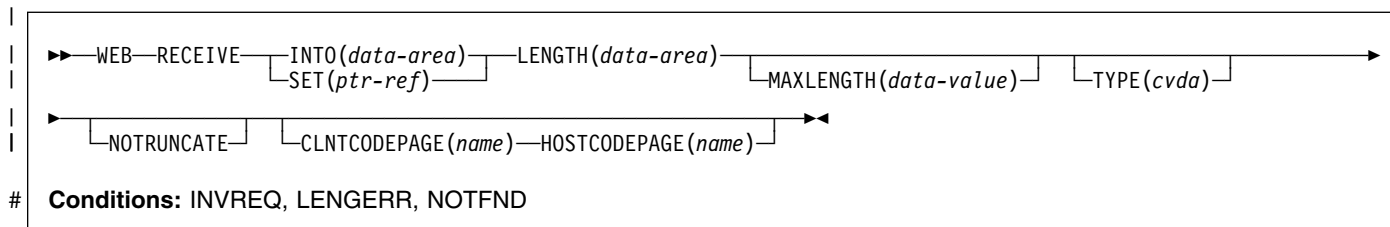
6 A header has been found which is not in the format NAME:VALUE.

LENGERR
 RESP2 values are:

- 1** NAMELENGTH or VALUELENGTH is less than or equal to zero.
- 4** The header name has been truncated during a browse operation because the receiving buffer is too small.
- 5** The header value has been truncated because the receiving buffer is too small.

WEB RECEIVE

Receive an HTTP request.



Description

WEB RECEIVE receives the body of an HTTP Request (or all the data received for a non-HTTP Request) from the CICS Web interface or Business Logic Interface into an application-supplied buffer.

Options

CLNTCODEPAGE(name)

specifies the 40-character name of the codepage to be used when data is converted from the client codepage. If this is not specified, data is not converted, and is passed to the application in the form in which it was received over the socket connection. If you specify CLNTCODEPAGE you must also specify HOSTCODEPAGE. Valid values for CLNTCODEPAGE are the CICS-supported CLINTCP and SRVERCP values described in *CICS Family: Communicating from CICS on System/390*. Mapping is performed between the IANA Coded Character Set values used in HTTP requests to the CLINTCP values known to CICS; these values are listed in the *CICS Internet Guide*.

HOSTCODEPAGE(name)

specifies the 8-character name of the host codepage to be used when data is converted from the ASCII codepage in which it was received into the EBCDIC codepage in which CICS will process it. If this is not specified, data is not converted, and is passed to the application in the form in which it was received over the socket connection. If you specify HOSTCODEPAGE you must also specify CLNTCODEPAGE.

INTO(data-area)

specifies the buffer that is to contain the data being received.

LENGTH(data-area)

specifies a fullword binary data area which will be set to the length of the data received.

MAXLENGTH(data-value)

specifies the maximum amount, as a fullword binary value, of data that CICS is to recover, if INTO is specified.

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.

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.

SET(ptr-ref)

specifies a pointer reference that is to be set to the address of data received. The pointer reference is valid until the next receive command or the end of task.

TYPE(cvda)

returns a value of HTTPYES or HTTPNO to identify whether the data received is an HTTP or non-HTTP message.

Conditions

INVREQ

RESP2 values:

1 the command is issued in a non-CICS Web interface application.

LENGERR

occurs if data is discarded by CICS because its length exceeds the maximum the program accepts. RESP2 values:

1 The length is negative.

2 The length is positive and exceeds the program's maximum.

NOTFND

RESP2 values:

WEB RETRIEVE

7 occurs if either the HOSTCODEPAGE or
CLNTCODEPAGE parameter specifies an
invalid codepage.

WEB RETRIEVE

| Retrieve a document.

| **▶▶—WEB—RETRIEVE—DOCTOKEN(*data-area*)—◀◀**
| **Conditions:** INVREQ

| Description

| The WEB RETRIEVE command retrieves the DOCTOKEN of
| the document which was sent using an earlier WEB SEND
| command.

| Options

| **DOCTOKEN(*data-area*)**
| specifies a 16-byte buffer to contain the symbolic name
| of the document to be retrieved.

| Conditions

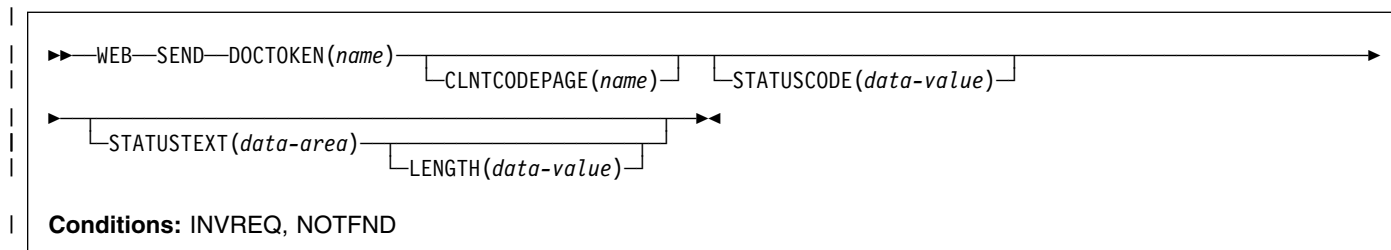
| INVREQ

| RESP2 values:

- | **1** The command is issued in a non-CICS Web
| interface application.
- | **2** A WEB SEND command has not been issued.

WEB SEND

Select a document for delivery.



Description

WEB SEND selects a document for delivery by the CICS Web interface or the Business Logic Interface.

Options

CLNTCODEPAGE(name)

specifies the 40-character name of the ASCII codepage in which the data is to be delivered to the browser. If this is not specified, data is not converted, and is sent to the browser in the form in which the application supplied it. Valid values for CLNTCODEPAGE are the CICS-supported CLINTCP and SRVERCP values described in *CICS Family: Communicating from CICS on System/390*. Mapping is performed between the IANA Coded Character Set values used in HTTP requests and the CLINTCP values known to CICS; these values are listed in the *CICS Internet Guide*.

DOCTOKEN(name)

specifies the 16-byte symbolic name of the document to be sent.

LENGTH(data-value)

specifies the length, as a fullword binary value, of the string supplied on the STATUSTEXT option.

STATUSCODE(data-value)

is a binary halfword value to be inserted on the status line. The code is intended to be a HTTP status code and should conform to the rules for HTTP status codes. However, CICS does not validate the code and it is the application's responsibility to ensure that the value is valid.

STATUSTEXT(data-area)

specifies a data-area containing text describing the value of the status code. The status text is intended to be a human-readable value for the status code.

Conditions

INVREQ

occurs for the following conditions. RESP2 values are:

- 1 The command is being issued in a non-CICS Web interface application.

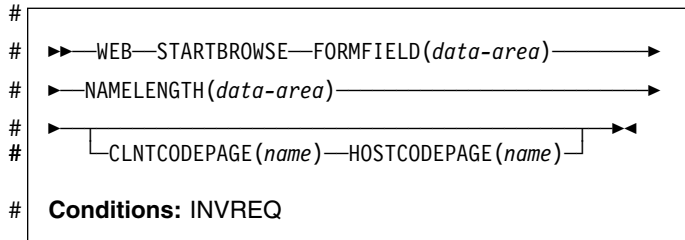
NOTFND

occurs for the following conditions. RESP2 values are:

- 1 The document has not been created or the name is incorrectly specified.
- 7 The specified CLNTCODEPAGE is not recognised or is incorrectly specified.

WEB STARTBROWSE FORMFIELD

Signal start of HTML form field browse.



Description

WEB STARTBROWSE FORMFIELD signals the start of a # browse of a set of name-value pairs in an HTML form that is # part of the body of an HTTP request being processed by the # current CICS task.

Options

CLNTCODEPAGE(name)

specifies the 40-character name of the codepage to be # used when data is converted from the client codepage. If # this is not specified, CICS obtains it from the charset # parameter on the Content-Type header of the HTTP # request for the current CICS task. If the Content-Type # header is not present, CICS used codepage 819 # (ISO-8859-1). If you specify CLNTCODEPAGE you # must also specify HOSTCODEPAGE. Valid values for # CLNTCODEPAGE are the CICS-supported CLINTCP # and SRVERCP values described in *CICS Family: # Communicating from CICS on System/390*. Mapping is # performed between the IANA Coded Character Set # values used in HTTP requests to the CLINTCP values # known to CICS; these values are listed in the *CICS # Internet Guide*.

FORMFIELD(data-area)

specifies the name of the form field at which browsing is # to start. It is a string of text containing the name of the # requested field. If a name is not specified, browsing # starts at the first name/value pair in the HTML form.

HOSTCODEPAGE(name)

specifies the 8-character name of the host codepage # used when data is converted from the ASCII codepage, # it was received in, into the EBCDIC, CICS processes it # in. If this is not specified the default (037) is used. If you # specify HOSTCODEPAGE you must also specify # CLNTCODEPAGE.

NAMELENGTH(data-value)

specifies the length, as a fullword binary value, of the # form field name. This field must be specified if # FORMFIELD is specified.

Conditions

INVREQ

occurs for the following conditions. RESP2 values are:

- # 1 The command is being issued in a non-CICS Web application.
- # 3 The command is being issued for a non-HTTP request.
- # 5 There is already a WEB STARTBROWSE in progress.
- # 11 The client codepage cannot be found.
- # 12 The server codepage cannot be found.
- # 13 No forms data has been supplied in the body of the HTTP request.
- # 14 The codepage combination for client and server is invalid.

LENGERR

occurs for the following conditions. RESP2 values are:

- # 1 NAMELENGTH or VALUELENGTH is less than or equal to zero.

NOTFND

occurs for the following conditions. RESP2 values are:

- # 1 The form field name given in the FORMFIELD parameter could not be found.

WEB STARTBROWSE HTTPHEADER

Signal start of HTTP header browse.

▶—WEB—STARTBROWSE—HTTPHEADER—▶◀

Conditions: INVREQ

Description

WEB STARTBROWSE HTTPHEADER signals the start of a browse of the HTTP header information.

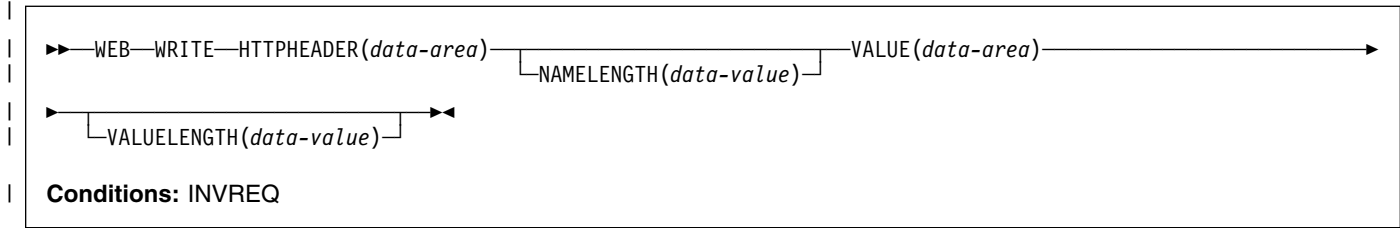
Conditions**INVREQ**

occurs for the following conditions. RESP2 values are:

- 1** The command is being issued in a non-CICS Web interface application.
- 3** The command is being issued for a non-HTTP request.

WEB WRITE

Build HTTP header information.

**Description**

WEB WRITE allows the application to add HTTP header information to the response.

Options**HTTPHEADER(data-area)**

specifies the name of the HTTP header to build. It is a string of text containing the name of the header. The name should conform to the standards specified in the current RFC for the HTTP protocol. No validation of the information is done by CICS. If the header already exists, it is replaced with the new information.

NAMELENGTH(data-value)

specifies the length, as a fullword binary value, of the HTTP header name.

VALUE(data-area)

specifies the value of the HTTP header. It is a text string containing the value of the header.

VALUELENGTH(data-area)

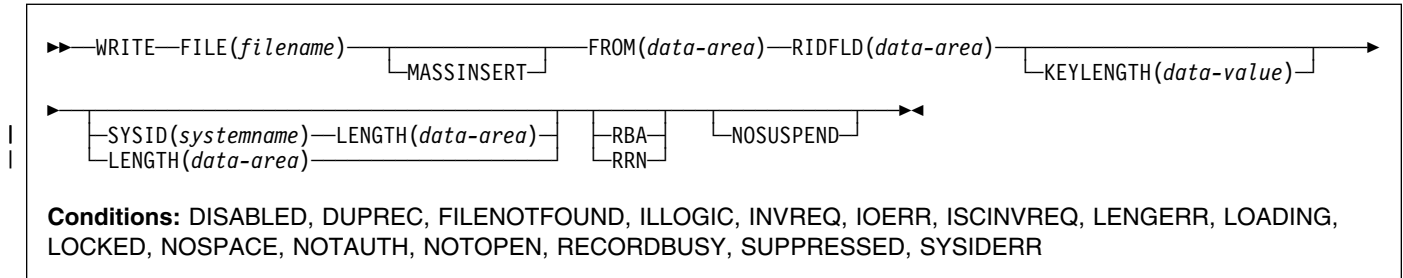
specifies the length, as a fullword binary value, of the HTTP header value.

Conditions**INVREQ**

The command is being issued in a non-CICS Web interface environment.

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).

| When this command is used to write a record to a coupling
| facility data table, the update is made to the data table in the
| coupling facility (unless it is 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.

WRITE

Note: The only data sets greater than 4GB (that is, defined
with the extended format / extended addressability
attribute) supported by CICS are KSDS, and then
only if they are accessed by key. CICS does not
support ESDS or RRDS data sets defined with the
extended attribute.

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.

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-area*)

| specifies the length, as a halfword binary value, of the
| data area where the record is written from.

| This option must be specified if SYSID is specified.

| If the file is on a remote system, the LENGTH parameter
| need not be set here but must be set in the file resource
| definition.

| If the file is on a local system, the LENGTH parameter
| must be set for variable-length records, using the INTO
| option, but not for fixed-length records. It is, however,
| advisable to specify the length for fixed-length records
| 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 or coupling facility data tables.

NOSUSPEND (RLS only)

The request does not wait if VSAM is holding an active lock against the record, including records locked as the result of a DEADLOCK.

A task could wait when it issues a WRITE request if the key is for a record that is being modified, created, or

deleted by another task, because VSAM always acquires the lock first.

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. You cannot use RBA for data sets that are greater than 4GB in size.

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 BDAM 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 defaulted from the FROM option using the length attribute reference in assembler language, or STG and CSTG in PL/I. LENGTH must be specified explicitly in OS/VS COBOL or C.

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.

This condition is also raised for a coupling facility data table that uses the contention model, even if another task has read the record with the same key for update. (For a coupling facility data table that uses the locking model, and for all other kinds of files, if another task has read the record for update, it is locked, and the WRITE request waits for the lock to be released, rather than returning a DUPREC response immediately.)

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 377 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 BDAM file.
- 40** A BDAM key conversion error occurred.
- 44** The WRITE command does not conform to the format of WRITE for a user-maintained or coupling facility data table (for example, MASSINSERT or RBA is specified).
- 51** A WRITE command specifying the RBA keyword was issued against a KSDS file that is being accessed in RLS mode. RLS mode does not support relative byte address access to KSDS files.

56 An attempt to update a recoverable coupling facility data table has failed because the current unit of work has already updated 1024 recoverable coupling facility data tables. You cannot update more than 1024 recoverable coupling facility data tables within a unit of work

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 377 for details.)

For VSAM files, IOERR normally indicates a hardware error.

For BDAM files, IOERR could mean that you are trying to write to a BDAM track address that is not defined for the data set.

For a coupling facility data table, an IOERR indicates a bad response returned from a coupling facility access.

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:

- 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 BDAM file with undefined format records.
- 14** An incorrect length is specified for a write operation involving fixed-length records.

LOADING

RESP2 values:

- 104** The request cannot be satisfied because it is issued against a data table that is still being loaded. The condition can be raised for one of the following reasons:
 - The WRITE specifies a record key that has out of range of the records so far loaded into a coupling facility data table. Records can be added while a CFDT is loading only if the

WRITE

specified key is within the range of those records already loaded.

The **LOADING** response can also be returned for a coupling facility data table that has failed during loading. For more information about what happens if the load for a coupling facility data table fails, see the description of the **XDTLC** global user exit in the *CICS Customization Guide*.

- 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.

If your application programs encounter the **LOADING** condition persistently or too frequently, check that this is not caused by conflicting file definitions that reference the same data set.

Default action: terminate the task abnormally.

LOCKED

RESP2 values:

- 106** An attempt has been made to write a record, but a retained lock exists against the key of this record.

Default action: abend the task with code AEX8.

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 or coupling facility data table has already been reached.

This condition can also occur for a recoverable coupling facility data table when the table apparently contains fewer than the maximum number of records allowed if there are uncommitted updates outstanding.

- 103** CICS is unable to get sufficient storage in the CICS address space to create an in-memory table entry for the record being written.

- 108** There is insufficient space in the coupling facility data table pool to store the record.

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** **NOTOPEN (RESP2 60)** is returned for one of the following reasons:

- 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. You can also make **CLOSED**, **UNENABLED** the initial state, by specifying **STATUS(UNENABLED)** and **OPENTIME(FIRSTREF)** on the **FILE** resource definition. (For **BDAM** files, you use the **FILSTAT** parameter of the **DFHFCT TYPE=FILE** macro.)
- The requested file is **OPEN** and in use by other transactions, but a **CLOSE** request against the file has been received.
- A **WRITE** request is issued against a data set is quiesced, or is being quiesced, as a result of a **SET DSNNAME QUIESCED** or **IMMQUIESCED** command.

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.

RECORDBUSY

RESP2 values:

- 107** **NOSUSPEND** is specified on the request but **VSAM** holds an active lock against the record, which would cause the request to wait (see "Retained and active locks" below).

Default action: abend the task with code AEX9.

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 or coupling facility data table.

Default action: terminate the task abnormally.

SYSIDERR

RESP2 values:

- 130** The **SYSID** option specifies a name that is neither the local CICS region nor a remote system defined to CICS by a **CONNECTION** definition. **SYSIDERR** also occurs when the link to the remote system is closed.

- 131** For a coupling facility data table, the connection to the coupling facility data table server has failed. This could be because the server itself has failed, or the server is available, but CICS has failed to connect to it.

| **132** The WRITE is issued against a coupling facility
| data table that no longer exists, probably because
| of a coupling facility failure, in which case the
| coupling facility data table server also fails. See
| the *CICS System Definition Guide* for information
| about restarting a coupling facility data table
| server and reloading a table.

Default action: terminate the task abnormally.

Retained and active locks: RECORDBUSY refers to active locks, and LOCKED refers to retained locks:

- READNEXT requests for records that have *retained* locks are always rejected with a LOCKED response.
- READNEXT requests for records that have *active* locks wait for the lock to be released, except when the NOSUSPEND keyword is specified, in which case CICS returns the RECORDBUSY response.

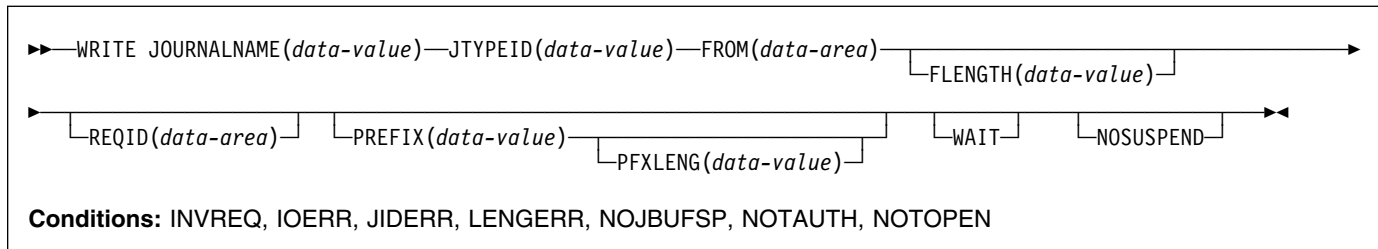
Examples

For example:

```
EXEC CICS WRITE  
FROM(RECORD)  
LENGTH(DATLEN)  
FILE('MASTER')  
RIDFLD(KEYFLD)
```

WRITE JOURNALNAME

Create a journal record



Description

WRITE JOURNALNAME writes a journal record from the specified data area to the system logger log stream that corresponds to the CICS journal name, or to SMF. The request can be for synchronous or asynchronous output; definitions of these terms, and information regarding the synchronization of journal output, are in the *CICS Application Programming Guide*.

Options

LENGTH(data-value)

specifies, as a full word binary value, the length in bytes of the user data to be built into the journal record.

Note that the maximum total length of a journal record depends on a number of factors:

- There is a limit of 32KB minus 400 bytes if the journal is using SMF.
- The limit for journals that map to log streams is the value expressed in the MAXBUFSIZE attribute for the structure being used minus 400 bytes. This has to include the user data, the prefix data, and the 2-byte JTYPEID.

Note: Data longer than 32K bytes cannot be read by offline jobs using the SUBSYS=LOGR interface.

FROM(data-area)

specifies the user data to be built into the journal record.

JOURNALNAME(data-value)

specifies a 1- to 8-character journal name. The valid characters for a journal name are the upper-case letters A through Z, the numeric characters 0 through 9, and the special symbols ¢ @ and #.

On first reference to this journal name, CICS must be able to map the journal name to a corresponding MVS system loggerlog stream, or MVS SMF data set. To do this, CICS searches the installed JOURNALMODEL definitions, looking for a matching journal name in a journal model. CICS looks for either a specific match or

a generic match. If a matching entry cannot be found, CICS attempts to use a default log stream name.

To write to the CICS system log, specify DFHLOG as the journal name.

Note: The CICS system log should be used only for short-lived data required for recovery purposes. You should not write user records for such things as audit trails to it.

To write to journals defined using the journal numbering convention (for example, to the auto journals defined in file resource definitions), specify the name as DFHJnn, where nn is the journal number in the range 1 to 99.

Note: Specifying DFHJ01 on this command refers to a user journal, *not* the system log.

You cannot write to a forward recovery log that is known to CICS only by its 26-character log stream name (as derived directly from the VSAM ICF catalog) unless you write to a journal whose matching JOURNALMODEL is associated with the same log stream name.

JTYPEID(data-value)

specifies a 2-character identifier to be placed in the journal record to identify its origin.

NOSUSPEND

specifies that the application program is not to be suspended for the NOJBUFSP condition. The user record is ignored.

PFXLENG(data-value)

specifies the length (halfword binary value) in bytes of the user prefix data to be included in the journal record.

Note that the maximum total length of a journal record depends on a number of factors:

- There is a limit of 32KB minus 400 bytes if the journal is using SMF.
- The limit for journals that map to log streams is the value expressed in the MAXBUFSIZE attribute for the structure being used minus 400 bytes. This has

to include the prefix data, the user data, and the 2-byte JTYPEID.

The minimum value is 0. See FLENGTH for the limits to the size of a journal record.

Note: Data longer than 32K bytes cannot be read by offline jobs using the SUBSYS=LOGR interface.

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 token that can be used for synchronization. REQID is only valid for asynchronous output (that is, the WAIT option is not specified).

WAIT

specifies that synchronous journal output is required. The requesting task waits until the record has been hardened.

Conditions

INVREQ

the command is not valid for processing by CICS.

Default action: Terminate the task abnormally.

IOERR

a journal record has not been output because an irrecoverable error condition was returned by the system logger log stream or by SMF.

Default action: If the log is the system log, CICS either quiesces or abends CICS. If the log is a general log, the task is terminated abnormally.

JIDERR

CICS cannot connect to a log stream referenced by the specified journal name, for one of the following reasons:

- The log stream does not exist and cannot be created dynamically using default model definitions.
- The log stream is a DASD-only log stream to which a CICS region in another MVS image is currently connected.

Default action: terminate the task abnormally.

LENGERR

the aggregate length of the journal record, comprising the user data (FROM, JTYPE, and PREFIX data) and the CICS header data, is too great to fit the maximum block size allowed for the log stream.

Default action: terminate the task abnormally.

NOJBUFSP

the journal buffers are logically full (that is, the current buffer has insufficient space for this journal record, and I/O is in progress on the alternate buffer).

Default action: CICS suspends 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. (You can override the default action by the NOSUSPEND option.)

NOTAUTH

a resource security check has failed on the JOURNALNAME(*data-value*). resource.

Default action: terminate the task abnormally.

NOTOPEN

occurs in any of the following situations:

- The command cannot be executed because the specified journal has been explicitly disabled by the user.
- The request could not be satisfied because the specified journal was defined using a journal model that maps it onto the logstream that is being used as the system log for this CICS system. The error is detected when trying to connect to the logstream and results in a definition for the JOURNALNAME being installed and set to 'failed'.

Default action: terminate the task abnormally.

Examples

The following example shows how to write synchronous journal output and wait for the output operation to be completed:

```
EXEC CICS WRITE
      JOURNALNAME('ACCTSJNL')
      JTYPEID('XX')
      FROM(KEYDATA)
      FLENGTH(40000)
      PREFIX(PROGNAME)
      PFXLENG(6)
      WAIT
```

The following example shows how to write deferred (asynchronous) user recovery data to the CICS system log:

```
EXEC CICS WRITE
      JOURNALNAME('DFHLOG')
      JTYPEID('UR')
      FROM(COMDATA)
      FLENGTH(10)
      REQID(ENTRYID)
```

WRITE JOURNALNUM

WRITE JOURNALNUM

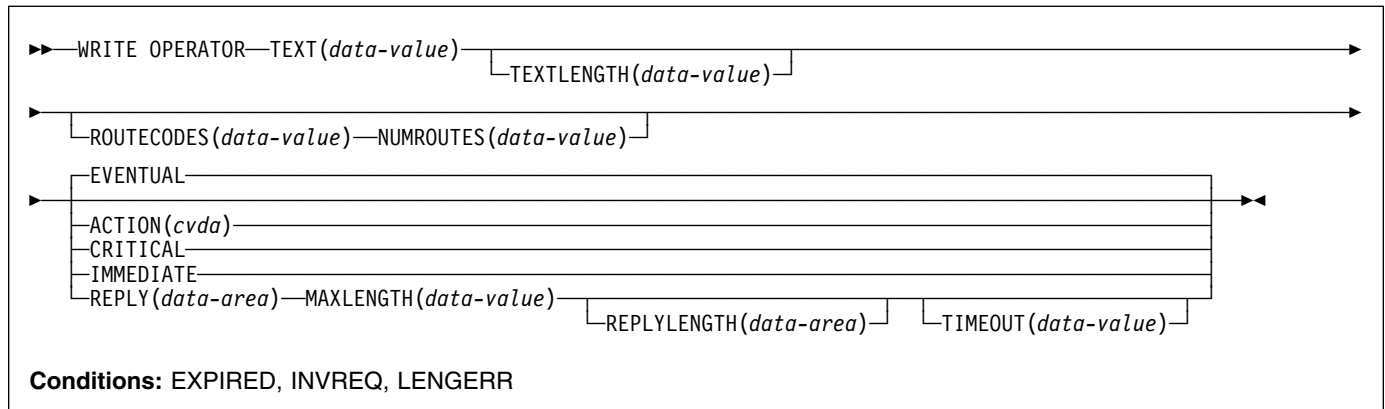
Create a journal record.

Note: This command is supported for compatibility with earlier releases of CICS. It is superseded by the `WRITE JOURNALNAME` command, which you are recommended to use instead.

The syntax is, apart from `FLENGTH`, the same as for `WRITE JOURNALNAME` except that `JOURNALNUM` specifies a numeric instead of a character value. The numeric value, `nn`, is in the range 01–99 and corresponds to journal name **DFHJnn**.

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).

Retained messages can be handled by the console operator in a variety of ways (see the *MVS/ESA Operations: System Commands* manual). 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, but is still retained by the operating system until the operator explicitly deletes it.

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

WRITE OPERATOR

either a reply is received or the TIMEOUT period expires.

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.

| For COBOL programs to be compiled with a Language
| Environment-conforming compiler, and translated with
| the COBOL3 translator option, there is a restriction in
| the length of data-value, which cannot exceed 160
| bytes. If you are using an earlier compiler, such as VS
| COBOL II, that does not support the COBOL3 translator
| option, you must use a data-area that contains the text
| to be sent to the operator, and not 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, LOCKED, 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 RECORDSIZE and associated formations specified in the TDQUEUE resource definition. The check is made after the XTDOU 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 XTDOU exit has been invoked.

Default action: terminate the task abnormally.

LOCKED

occurs when the request cannot be performed because use of the queue has been restricted owing to a unit of work failing in-doubt. This can happen on any request for a logically-recoverable queue defined with WAIT(YES) and WAITACTION(REJECT) in the TDQUEUE resource definition.

Specify WAIT(YES) and WAITACTION(QUEUE) in the TDQUEUE resource definition if you want the transaction to wait.

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.

WRITEQ TD

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.

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.

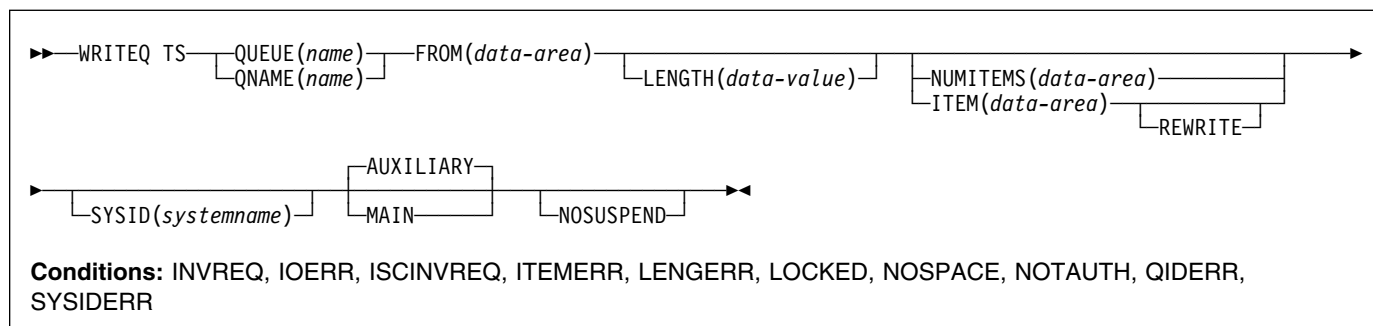
Examples

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 is the default value for the first write.

This option is ignored:

- for an existing queue,
- if a TSMODEL resource definition with a matching prefix is installed in the system.
- if the AUXILIARY option is specified for a temporary storage data queue that resides in a temporary storage pool.

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) 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.

The maximum length is 32763. 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 a TSMODEL resource definition with a matching prefix is installed in the system.

WRITEQ TS

- if the MAIN option is specified for a temporary storage data sharing queue that resides in a temporary storage pool.

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 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 the MAIN queue.

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.

QNAME(*name*)

an alternative to QUEUE, QNAME specifies the symbolic name (1–16 characters) of the queue to be written to. If the queue name appears in the TST, and the entry is marked as remote, or if the QNAME is described by a TSMODEL resource definition which identifies a remote system, 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 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 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

RESP2 values:

5 There is an irrecoverable input/output error for a shared queue.

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 in any of the following situations:

- The length of the stored data is zero or negative.
- The length of the stored data is greater than 32763.

Default action: terminate the task abnormally.

LOCKED

RESP2 values:

0 The request cannot be performed because use
 # of the queue has been restricted owing to a unit
 # of work failing in-doubt.

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

RESP2 values:

101 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
- Temporary storage pool

Default action: terminate the task abnormally.

SYSIDERR

RESP2 values:

4 The CICS region in which the temporary
 # storage command is executed fails to connect
 # to the TS server managing the TS pool that
 # supports the referenced temporary storage
 # queue. (For example, this can happen if the
 # CICS region is not authorized to access the
 # temporary storage server).

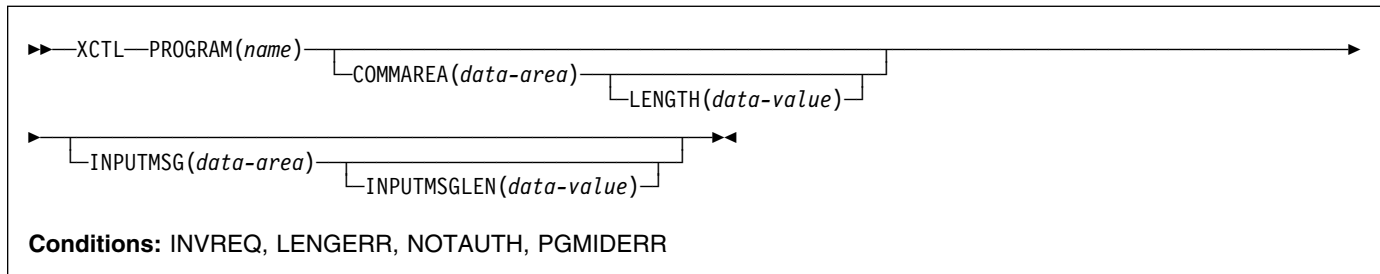
SYSIDERR can also occur if the temporary
 # storage server has not been started, or
 # because the server has failed (or been stopped)
 # while CICS continues executing. Also occurs in
 # any of the following situations:

- # • 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.

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.
- #**
- #**
- #**
- 43** An XCTL has been attempted to a hot-pooled Java program object while there is already a hot-pooled program on the stack.
- 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*).

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 the DFHRPL concatenation, 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
```

EIBCPOSN

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 eibcposn;
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.

```
COBOL: PIC S9(7) COMP-3.
C:      char eibdate [4];
PL/I:   FIXED DEC(7,0)
ASM:    PL4
```

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
```

EIBERR • EIBFN

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 TD
```

```
0A02 WRITEQ TS  
0A04 READQ TS  
0A06 DELETEQ TS
```

```
0C02 GETMAIN  
0C04 FREEMAIN
```

0E02 LINK		2014 DEFINE DCOUNTER
0E04 XCTL		2016 GET DCOUNTER
0E06 LOAD		2018 UPDATE DCOUNTER
0E08 RETURN		201A DELETE DCOUNTER
0E0A RELEASE		201C REWIND DCOUNTER
0E0C ABEND		201E QUERY DCOUNTER
0E0E HANDLE ABEND		
		3402 DEFINE ACTIVITY
1002 ASKTIME		3404 DEFINE PROCESS
1004 DELAY		3406 RUN ACTIVITY
1006 POST		3408 RUN ACQPROCESS
1008 START		340E ACQUIRE PROCESS
1008 START ATTACH		3410 ACQUIRE ACTIVITYID
1008 START BREXIT		3412 DELETE CONTAINER
100A RETRIEVE		3414 GET CONTAINER
100C CANCEL		3416 PUT CONTAINER
		3418 RESET ACTIVITY
1202 WAIT EVENT		341A CHECK ACTIVITY
1204 ENQ		341C CANCEL ACTIVITY
1206 DEQ		341E CANCEL ACQPROCESS
1208 SUSPEND		3420 SUSPEND ACTIVITY
		3422 SUSPEND ACQPROCESS
1402 WRITE JOURNALNUM		3424 RESUME ACTIVITY
1404 WAIT JOURNALNUM		3426 RESUME ACQPROCESS
1406 WRITE JOURNALNAME		3428 DELETE ACTIVITY
1408 WAIT JOURNALNAME		342A LINK ACQPROCESS
		342C LINK ACTIVITY
1602 SYNCPOINT		342E CANCEL ACQACTIVITY
		3430 RUN ACQACTIVITY
1802 RECEIVE MAP		3432 LINK ACQACTIVITY
1804 SEND MAP		3434 SUSPEND ACQACTIVITY
1806 SEND TEXT		3436 RESUME ACQACTIVITY
1808 SEND PAGE		3438 CHECK ACQPROCESS
180A PURGE MESSAGE		343A CHECK ACQACTIVITY
180C ROUTE		343C RESET ACQPROCESS
180E RECEIVE PARTN		
1810 SEND PARTNSET		3602 DEFINE INPUT EVENT
1812 SEND CONTROL		3602 DEFINE COMPOSITE EVENT
		3604 DELETE EVENT
1C02 DUMP		3608 ADD SUBEVENT
		360A REMOVE SUBEVENT
1E02 ISSUE ADD		360E TEST EVENT
1E04 ISSUE ERASE		3610 RETRIEVE REATTACH EVENT
1E06 ISSUE REPLACE		3612 RETRIEVE SUBEVENT
1E08 ISSUE ABORT		3614 DEFINE TIMER
1E0A ISSUE QUERY		3616 DELETE TIMER
1E0C ISSUE END		3618 CHECK TIMER
1E0E ISSUE RECEIVE		361A FORCE TIMER
1E10 ISSUE NOTE		
1E12 ISSUE WAIT		3802 WEB RECEIVE
1E14 ISSUE SEND		3804 WEB SEND
		3806 WEB READ
2002 BIF DEEDIT		3808 WEB STARTBROWSE
		380A WEB READNEXT
2004 DEFINE COUNTER		380C WEB ENDBROWSE HTTPHEADER
2006 GET COUNTER		380E WEB WRITE
2008 UPDATE COUNTER		3810 WEB EXTRACT
200A DELETE COUNTER		3814 WEB RETRIEVE
200C REWIND COUNTER		
200E QUERY COUNTER		3C02 DOCUMENT CREATE

EIBFREE • EIBRCODE

I 3C04 DOCUMENT INSERT
 I 3C06 DOCUMENT RETRIEVE
 I 3C08 DOCUMENT SET

I 3E0E EXTRACT TCP/IP
 I 3E10 EXTRACT CERTIFICATE

4802 ENTER TRACENUM
 4804 MONITOR

4A02 ASKTIME ABSTIME
 4A04 FORMATTIME

5602 SPOOL/OPEN
 5604 SPOOL/READ
 5606 SPOOL/WRITE
 5610 SPOOL/CLOSE

5E06 CHANGE TASK
 5E22 WAIT EXTERNAL
 5E32 WAITCICS

6A02 QUERY SECURITY
 6C02 WRITE OPERATOR
 6C12 ISSUE DFHWTO

7402 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 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 eibrcode [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	EIBFN	EIBRCODE	Condition
04 ..	EB	CBIDERR	10 ..	81	NOTFND
04 ..	EC	PARTNERIDERR	10 ..	D0	SYSIDERR (see note 1)
04 ..	ED	NETNAMEIDERR	10 ..	D1	ISCINVREQ
04 ..	F1	TERMERR	10 ..	D6	NOTAUTH
04 20	EOC	10 ..	D8	USERIDERR
04 40	INBFMH	10 ..	E1	LENGERR
04 F6	NOSTART	10 ..	E9	ENVDEFERR
04 F7	NONVAL	10 ..	FF	INVREQ
06 ..	01	FILENOTFOUND	12 ..	32	ENQBUSY
06 ..	02	ILLOGIC (see note 6)	12 ..	E0	INVREQ
06 ..	03	LOCKED	12 ..	E1	LENGERR
06 ..	05	RECORDBUSY	14 ..	01	JIDERR
06 ..	08	INVREQ	14 ..	02	INVREQ
06 ..	0C	NOTOPEN	14 ..	05	NOTOPEN
06 ..	0D	DISABLED	14 ..	06	LENGERR
06 ..	0F	ENDFILE	14 ..	07	IOERR
06 ..	80	IOERR (see note 6)	14 ..	09	NOJBUFSP
06 ..	81	NOTFND	14 ..	D6	NOTAUTH
06 ..	82	DUPREC	16 ..	01	ROLLEDBACK
06 ..	83	NOSPACE	18 ..	01	INVREQ
06 ..	84	DUPKEY	18 ..	02	RETPAGE
06 ..	85	SUPPRESSED	18 ..	04	MAPFAIL
06 ..	86	LOADING	18 ..	08	INVMPSZ (see note 7)
06 ..	D0	SYSIDERR (see note 1)	18 ..	20	INVERRTERM
06 ..	D1	ISCINVREQ	18 ..	40	RTESOME
06 ..	D6	NOTAUTH	18 ..	80	RTEFAIL
06 ..	E1	LENGERR	18 ..	E1	LENGERR
08 ..	01	QZERO	18 ..	E3	WRBRK
08 ..	02	QIDERR	18 ..	E4	RDATT
08 ..	04	IOERR	18 02	PARTNFAIL
08 ..	08	NOTOPEN	18 04	INVPARTN
08 ..	10	NOSPACE	18 08	INVPARTNSET
08 ..	C0	QBUSY	18 10	INVLDC
08 ..	D0	SYSIDERR (see note 1)	18 20	UNEXPIN
08 ..	D1	ISCINVREQ	18 40	IGREQCD
08 ..	D6	NOTAUTH	18 80	TSIOERR
08 ..	D7	DISABLED	18 01	OVERFLOW
08 ..	E0	INVREQ	18 04	EODS
08 ..	E1	LENGERR	18 08	EOC
0A ..	01	ITEMERR	18 10	IGREQID
0A ..	02	QIDERR	1A ..	E0	INVREQ
0A ..	04	IOERR	1A ..	04	DSSTAT
0A ..	08	NOSPACE	1A ..	08	FUNCERR
0A ..	20	INVREQ	1A ..	0C	SELNERR
0A ..	D0	SYSIDERR (see note 1)	1A ..	10	UNEXPIN
0A ..	D1	ISCINVREQ	1A ..	E1	LENGERR
0A ..	D6	NOTAUTH	1A 11	EODS
0A ..	E1	LENGERR	1A 2B	IGREQCD
0C ..	E1	LENGERR	1A 20	EOC
0C ..	E2	NOSTG	22 ..	80	INVEXITREQ
0E ..	01	PGMIDERR	4A 01	INVREQ
0E ..	D6	NOTAUTH	56 0D	NOTFND
0E ..	E0	INVREQ	56 10	INVREQ
0E ..	E1	LENGERR	56 13	NOTOPEN
0E ..	F1	TERMERR	56 14	ENDFILE
10 ..	01	ENDDATA	56 15	ILLOGIC
10 ..	04	IOERR	56 16	LENGERR
10 ..	11	TRANSIDERR	56 2A	NOSTG
10 ..	12	TERMIDERR	56 46	NOTAUTH
10 ..	20	EXPIRED			

EIBRCODE

EIBFN	EIBRCODE	Condition
56 50 ...	NOSPOOL
56 55 ...	ALLOCERR
56 56 ...	STRELERR
56 57 ...	OPENERR
56 58 ...	SPOLBUSY
56 59 ...	SPOLEERR
56 5A ...	NODEIDERR

Notes:

- 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 found4.
.. 04 0C .. .. . modename not valid4.
.. 04 10 .. .. . task canceled or timed
                    out during allocation4.
.. 04 14 .. .. . mode group is out of
                    service4.
.. 04 18 .. .. . close - DRAIN=ALL4.
.. 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 *CICS Intercommunication Guide*.

- 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.

```

Further information about SESSIONERR can be found in the *CICS Intercommunication Guide*.

- 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:

```

.. .. . 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 *CICS Intercommunication Guide*.

- When INVREQ occurs during terminal control operations, further information is provided in bytes 1 or 3 of EIBRCODE as follows:

⁴ For APPC only.

.. 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 xx xx ..	BDAM response.
.. xx	VSAM return code.
.. .. xx	VSAM error code.

where:

byte 3 = VSAM problem determination code (ILLOGIC only)
byte 4 = VSAM component code (ILLOGIC only)

Details of these response codes are given in the *DFSMS Macro Instructions for Data Sets* manual for VSAM, and the *DFSMS/MVS V1.3 Using Data Sets (SC26-4922)* for BDAM.

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 27 on page 437.

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

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	55	ENQBUSY
01	ERROR	56	ENVDEFERR

No.	Condition	No.	Condition
02	RDATT	57	IGREQCD
03	WRBRK	58	SESSIONERR
04	EOF	59	SYSBUSY
05	EODS	60	SESSBUSY
06	EOC	61	NOTALLOC
07	INBFMH	62	CBIDERR
08	ENDINPT	63	INVEXITREQ
09	NONVAL	64	INVPARTNSET
10	NOSTART	65	INVPARTN
11	TERMIDERR	66	PARTNFAIL
12	FILENOTFOUND	69	USERIDERR
13	NOTFND	70	NOTAUTH
14	DUPREC	72	SUPPRESSED
15	DUPKEY	80	NOSPOOL
16	INVREQ	81	TERMERR
17	IOERR	82	ROLLEDBACK
18	NOSPACE	83	END
19	NOTOPEN	84	DISABLED
20	ENDFILE	85	ALLOCERR
21	ILLOGIC	86	STRELERR
22	LENGERR	87	OPENERR
23	QZERO	88	SPOLBUSY
24	SIGNAL	89	SPOLEERR
25	QBUSY	90	NODEIDERR
26	ITEMERR	91	TASKIDERR
27	PGMIDERR	92	TCIDERR
28	TRANSIDERR	93	DSNNOTFOUND
29	ENDDATA	94	LOADING
31	EXPIRED	95	MODELIDERR
32	RETPAGE	96	OUTDESCRERR
33	RTEFAIL	97	PARTNERIDERR
34	RTESOME	98	PROFILEIDERR
35	TSIOERR	99	NETNAMERR
36	MAPFAIL	100	LOCKED
37	INVERRTERM	101	RECORDBUSY
38	INVMP5Z	102	UOWNNOTFOUND
39	IGREQID	103	UOWLNOTFOUND
40	OVERFLOW	104	LINKABEND
41	INVLDC	105	CHANGED
42	NOSTG	106	PROCESSBUSY
43	JIDERR	107	ACTIVITYBUSY
44	QIDERR	108	PROCESSERR
45	NOJBUFSP	109	ACTIVITYERR
46	DSSTAT	110	CONTAINERERR
47	SELNERR	111	EVENTERR
48	FUNCERR	112	TOKENERR
49	UNEXPIN	113	NOTFINISHED
50	NOPASSBKRD	114	POOLERR
51	NOPASSBKWR	115	TIMERERR
53	SYSIDERR	116	SYMBOLERR
54	ISCINVREQ	117	TEMPLATERR

```

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	8
Interval control	Transaction name	4
Journal control	Journal number	H
	Journal name	8
Program control	Program name	8
Temporary storage control	TS queue name	8 or 16
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.

Identifiers greater than eight characters in length will be truncated.

```

COBOL: PIC X(8).
C:      char eibrsrce [8];
PL/I:   CHAR(8)
ASM:    CL8
    
```

EIBSIG
indicates that SIGNAL has been received (X'FF').

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 X(1).
C:      unsigned char eibsig;
PL/I:   CHAR(1)
ASM:    CL1

```

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

```

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 eibtaskn [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+), and can contain a
# value in the range 0000000+ to 0240000+. Both
# 0000000+ and 0240000+ are valid. .

```

```

# COBOL: PIC S9(7) COMP-3.
# C:      char eibtime [4];
# PL/I:   FIXED DEC(7,0)
# ASM:    PL4

```

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

```

EIBTRNID

contains the symbolic transaction identifier of the task.

```

COBOL: PIC X(4).
C:      char eibtrnid [4];
PL/I:   CHAR(4)
ASM:    CL4

```

EIBTRNID

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 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
..1.1	X'50'	TCAM
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 .1.1	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

Code		Meaning
1.11	X'B0'	SDLC device class
1.11 ...1	X'B1'	3601
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.

Use of 'modern' COBOL CALL syntax

For COBOL3, the 'by content' and 'by reference' phrases are used and each generated call is terminated by 'end-call'.

For example, the following EXEC statement:

```
EXEC CICS RETURN TRANSID('fred')
      COMMAREA(mycommarea) END-EXEC.
```

is currently translated to:

```
MOVE ' \      00027 ' TO DFHEIV0
MOVE 'fred' TO DFHC0040
MOVE LENGTH OF mycommarea TO DFHB0020
CALL 'DFHEI1' USING DFHEIV0 DFHC0040 mycommarea
                        DFHB0020.
```

For COBOL3, it is translated as:

```
Move length of mycommarea to dfhb0020
Call 'DFHEI1' using by content
      x'0e08e0000700001000f0f0f0f2f7404040'
      by content 'fred' by reference mycommarea
      by reference dfhb0020 end-call.
```

Note that argument zero is passed 'by content' as a hexadecimal literal.

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:

Translated code

```
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 7 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('DFHçAGA') MAPONLY ERASE	
END	
The above source program is translated to:	
DFHEIGBL ,	INSERTED BY TRANSLATOR
INSTRUCT CSECT	
DFHEIENT	INSERTED BY TRANSLATOR
* EXEC CICS SEND MAP('DFHçAGA') MAPONLY ERASE	
DFHECALL =X'1804C0000800000000046204000020',	
(CHAF7,=CL7'DFHçAGA*'),(____RF,DFHEIV00)	
DFHEIRET	INSERTED BY TRANSLATOR
DFHEISTG	INSERTED BY TRANSLATOR
DFHEIEND	INSERTED BY TRANSLATOR
END	

Figure 7. 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 8 shows a simple assembler-language application program that uses such variables in dynamic storage.

Source program	
DFHEISTG DSECT	
COPY DFHçAGA	INPUT MAP DSECT
COPY DFHçAGB	OUTPUT MAP DSECT
MESSAGE DS CL39	
INQUIRY CSECT	
EXEC CICS RECEIVE MAP('DFHçAGA')	
MVC NUMBO,KEYI	
MVC MESSAGE,=CL(L'MESSAGE)'THIS IS A MESSAGE'	
EXEC CICS SEND MAP('DFHçAGB') ERASE	
END	
The above source program is translated to:	
DFHEIGBL ,	INSERTED BY TRANSLATOR
DFHEISTG DSECT	
DFHEISTG	INSERTED BY TRANSLATOR
COPY DFHçAGA	INPUT MAP DSECT
COPY DFHçAGB	OUTPUT MAP DSECT
MESSAGE DS CL39	
INQUIRY CSECT	
DFHEIENT	INSERTED BY TRANSLATOR
* EXEC CICS RECEIVE MAP('DFHçAGA')	
DFHECALL =X'1802C0000800000000040900000020',	
(CHAF7,=CL7'DFHçAGA*'),(____RF,DFHçAGAI)	
MVC NUMBO,KEYI	
MVC MESSAGE,=CL(L'MESSAGE)'THIS IS A MESSAGE'	
* EXEC CICS SEND MAP('DFHçAGB') ERASE	
DFHECALL =X'1804C000080000000004E204000020',	
(CHAF7,=CL7'DFHçAGB*'),(____RF,DFHçAGBO)	
DFHEIRET	INSERTED BY TRANSLATOR
DFHEISTG	INSERTED BY TRANSLATOR
DFHEIEND	INSERTED BY TRANSLATOR
END	

Figure 8. 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 7 would become:

```
INSTRUCT DFHEIENT CODEREG=(2,3,4),  
          DATAREG=(13,5),  
          EIBREG=6  
EXEC CICS SEND  
MAP('DFHçAGA')  
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, control

Terminal 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 TCAM or 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.

TCAM-supported terminals and logical units

Because TCAM permits many applications to share a single network, the CICS-TCAM interface supports data streams rather than specific terminals or logical units.

Operations for terminals supported by TCAM use the same options as the terminals supported by other access methods. With the exception of the BUFFER option for the 3270, all options applicable for input operations are supported by CICS-TCAM. However, the conditions ENDINPT and EOF do not occur.

All output requests are the same for TCAM as for other CICS-supported terminals, except that:

- The ISSUE RESET command cannot be used
- The ISSUE COPY and ISSUE PRINT commands for the 3270 cannot be used
- The DEST option is available on CONVERSE and SEND commands, in addition to other appropriate options.

With the exception of 3650 logical units, operations for logical units supported by TCAM use the same options as logical units supported by VTAM.

BTAM programmable terminals

CICS Transaction Server for OS/390 Release 3 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.

When TCAM is used, all of this line control is handled by the message control facility rather than by CICS.

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 15.

Table 15. 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
SEND	Write initial (TI)	Read initial (TI)
	Write continue (TT)	Read continue (TT)
RECEIVE	Write reset (TR) ⁴	Read continue (TT)
	Read initial (TI)	Write initial (TI)
<ol style="list-style-type: none"> 1. 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. 2. 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). 3. The read continue is issued only if the EOT character is not received on the read interrupt. 4. 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.

Terminal control

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 22.

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 option SCRNSIZE, or in the TCT. 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 or by a printer supplied by the autoinstall user program, and it must have the correct buffer capacity. If the copy feature is also specified (COPY(YES) on 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.)

CICS supports only those SAA Resource Recovery return codes that match existing EXEC CICS commands. 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 16. 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 17 (Page 1 of 4). Summary of the CICS API by functional area</i>		
Functional area	Command	Supported?
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 17 (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 ASRASPC ASRASTG CMDSEC CWALENG INITPARM INITPARMLENG INVOKINGPROG NETNAME OPERKEYS OPID ORGABCODE PRINSYSID PROGRAM RESSEC RESTART RETURNPROG STARTCODE SYSID TASKPRIORITY TCTUALENG TRANPRIORITY TWALENG USERID	YES

<i>Table 17 (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 COMMAREA CWA EIB TCTUA TWA	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 JOURNALNAME WAIT JOURNALNAME WRITE JOURNALNUM WAIT JOURNALNUM	YES
Monitoring	MONITOR POINT	YES

API restrictions

<i>Table 17 (Page 4 of 4). Summary of the CICS API by functional area</i>		
Functional area	Command	Supported?
Program control	LINK LOAD RELEASE RETURN XCTL	YES
Note: LINK, RETURN and XCTL do not support INPUTMSG.		
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.

CVDA numeric values

Table 18. CVDAs and numeric values in alphabetic sequence

cvda	value
ALLOCATED	81
ALTERABLE	52
AUTOAUTH	1095
AUTOREGISTER	1094
BASESPACE	664
BASICAUTH	1092
CERTIFICAUTH	1093
CICSEXECKEY	381
CONFFREE	82
CONFRECEIVE	83
CONFSEND	84
CRITICAL	11
CTRLABLE	56
EVENTUAL	3
FREE	85
IMMEDIATE	2
LOG	54
NOAUTHENTIC	1091
NOLOG	55
NONCICS	661
NOTALTERABLE	53
NOTAPPLIC	1
NOTCTRLABLE	57
NOTPURGEABLE	161
NOTREADABLE	36
NOTUPDATABLE	38
PENDFREE	86
PENDRECEIVE	87
PURGEABLE	160
READABLE	35
RECEIVE	88
ROLLBACK	89
SEND	90
SUBSPACE	663
SYNCFREE	91
SYNCRECEIVE	92
SYNCSEND	93
TASK	233
UOW	246
UPDATABLE	37
USEREXECKEY	382

Table 19. 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	UOW
381	CICSEXECKEY
382	USEREXECKEY
661	NONCICS
663	SUBSPACE
664	BASESPACE
1091	NOAUTHENTIC
1092	BASICAUTH
1093	CERTIFICAUTH
1094	AUTOREGISTER
1095	AUTOAUTH

Appendix I. National language codes

Language codes are held as one character for NATLANG and NATLANGINUSE, and three characters for LANGUAGECODE and LANGINUSE.

Table 20. 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 21. 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 22 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 23 on page 413 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 22 (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 22 (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
DFHMFET	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 23. 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 24 on page 414 shows the meaning of each bit. The constants supplied in DFHMSRCA are listed in Table 25 on page 414.

Table 24. 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 25. 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.
DFHMSFIN	Input complete. Green on; emit short buzz; IN PROCESS mode reset.
DFHMSALR	Operator alert. Green, yellow, and red on; emit long buzz; IN PROCESS mode reset.
DFHMSALS	Operator alert. Green, yellow, and red on; emit long buzz; IN PROCESS mode set.
DFHMSIPY	IN PROCESS state set. Yellow on.
DFHMSIPN	IN PROCESS state reset.
DFHMSLKY	MSR operation inhibited. Yellow on.
DFHMSLKN	MSR input allowed. Green on. Yellow on.
DFHMSAEY	MSR autoenter on. Yellow on.
DFHMSAEN	MSR autoenter off. Yellow on.
DFHMSLBN	Long buzzer suppressed. Yellow on.
DFHMSLBYP	Long buzzer permitted. Yellow on.
DFHMSSBN	Short buzzer suppressed. Yellow on.
DFHMSSBY	Short buzzer permitted. Yellow on.
DFHMSNOP	Leave all MSR settings unchanged.

Attention identifier constants, DFHAID

The standard attention identifier list, DFHAID, simplifies testing the contents of the EIBAID field. Table 26 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 26. Standard list DFHAID</i>	
Constant	Meaning
DFHENTER	ENTER key.
DFHCLEAR	CLEAR key.
DFHPA1–DFHPA3	PA1–PA3 keys.
DFHPPF1–DFHPPF24	PF1–PF24 keys.
DFHOPID	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.

Before CICS can load a physical map, it requires an installed resource definition for the map object. You can either use program autoinstall to create the definition when the mapset is first used, or define a mapset in the CSD using the DEFINE MAPSET resource definition.

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.

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.

ADS descriptor

Physical maps produced by CICS Transaction Server for OS/390 Release 3 will also include an ADS descriptor in the output load module. This is provided to allow interpretation of the BMS Application Data Structure (the structure used by the application program for the data in SEND and RECEIVE MAP requests), without requiring your program to include the relevant DSECT or copybook at compile time.

The ADS descriptor contains a header with general information about the map, and a field descriptor for every field that appears in the ADS (corresponding to every named field in the map definition macro).

The ADS descriptor is generated for all maps. You can use the DSECT option to select the long form of the ADS, where all fields are aligned on 4-byte boundaries. The long form of the ADS is required by the 3270 Bridge when an interface to MQSeries is used.

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:

BMS macro summary



“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.

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 a physical map using an RDO 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, *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:

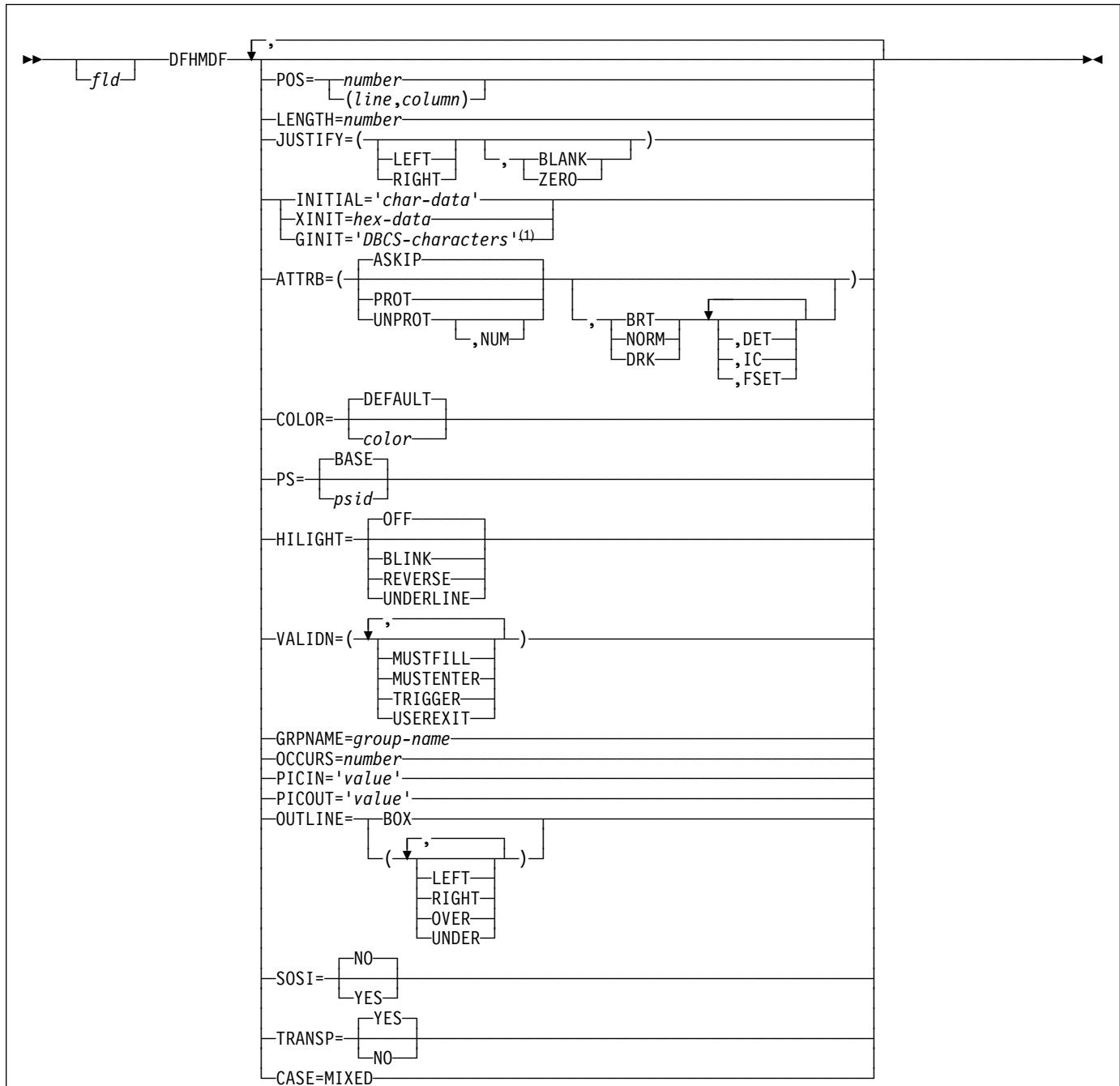
```

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

**Note:**

¹ DBCS characters start with a shift-out character X'0E' and end with a shift-in character X'0F'.

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 a physical map using an RDO transaction with the MAPSET attribute.

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 option COLOR.

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.

| The rules for defining groupnames are the same as for
| field names. See the *CICS Application Programming*
| *Guide* for details.

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.

HILIGHT

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 HILIGHT operand is ignored unless the terminal supports highlighting, as indicated by the RDO TYPETERM option HILIGHT(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 TCAM-supported 3600 and 3790, and 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.

In general LENGTH can be omitted if PICIN or PICOUT is specified, unless PICOUT defines a COBOL picture containing a currency symbol that will be replaced by a currency sign of length greater than 1. LENGTH 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

DFHMDF

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.

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 DFHMDS TYPE=DSECT,
      LANG=COBOL,
      MODE=INOUT
MAP DFHMDSI 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'
      DFHMDS 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.
```

Valid picture values for COBOL input maps are:

A P S V X 9 / and (

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.

Note: PL/I 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 currency string enclosed by less than (<) and greater than (>) symbols. For example:

```
DECLARE
    USPRICE PICTURE '$99.99',
    UKPRICE PICTURE '<£>99.99',
    EUPRICE PICTURE '<EUR>99.99';
```

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.

Valid picture values for COBOL output maps are:

A B E P S V X Z 0 9 , . + - ¢
CR DB / and (

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: PL/I 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 currency string enclosed by less than (<) and greater than (>) symbols. For example:

```
DECLARE
    USPRICE PICTURE '$99.99',
    UKPRICE PICTURE '<£>99.99',
    EUPRICE PICTURE '<EUR>99.99';
```

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 ECPRICE 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

DFHMDF

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 field can be processed by the BMS global user exits

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*.

USEREXIT

specifies that this field is to be processed by the BMS global user exits, XBMIN and XBMOU, if this field is received or transmitted in a 3270 datastream when the respective exit is enabled. For further information on the use of the BMS global user exits refer to the *CICS Customization Guide*.

The MUSTFILL, MUSTENTER and TRIGGER specifications are valid only for terminals that support the field validation extended attribute, otherwise they are ignored. The USEREXIT specification applies to all 3270 devices.

Note: The USEREXIT specification is totally unconnected with the field validation extended attribute as defined in the 3270 datastream architecture.

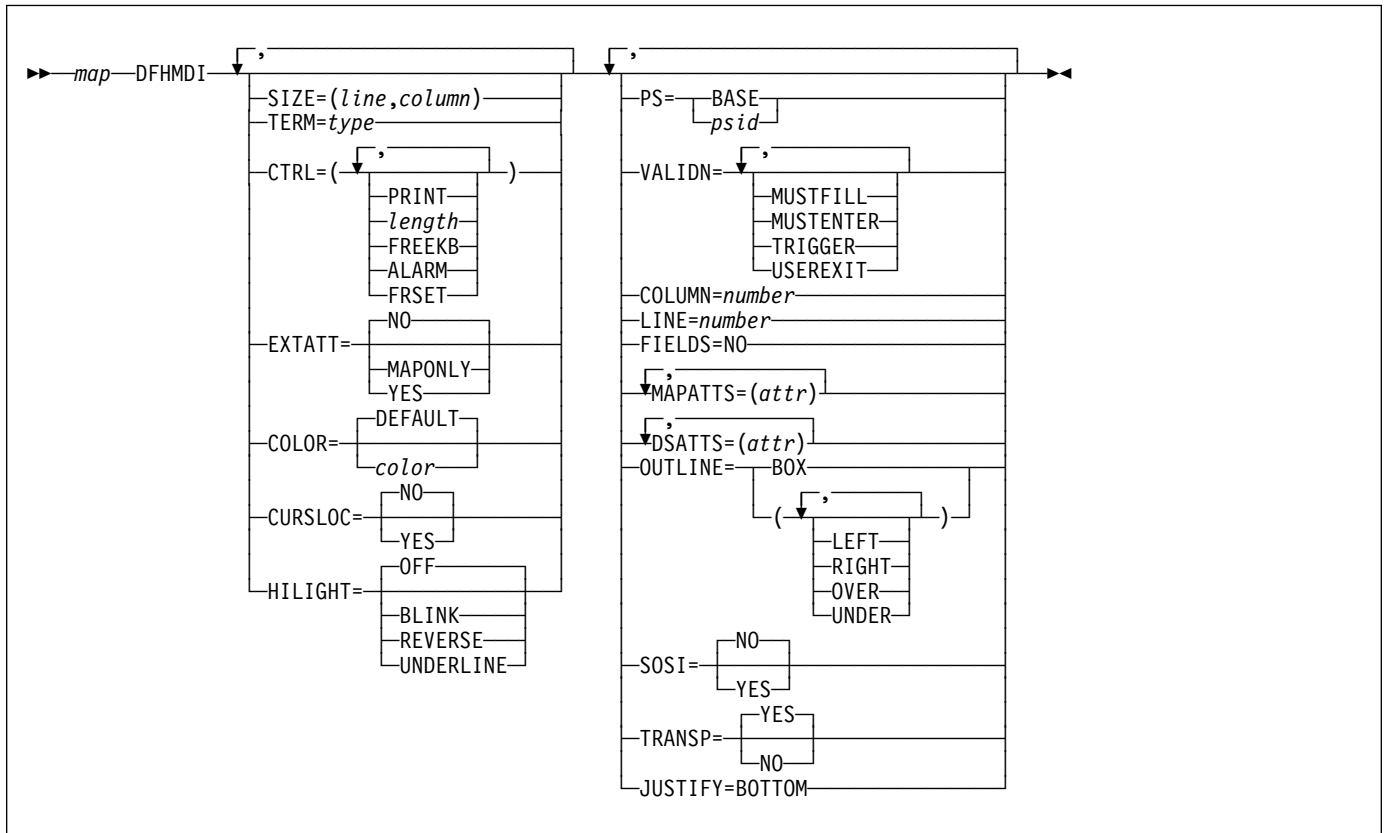
XINIT

see INITIAL, earlier in the list. Only one of GINIT, INITIAL, or XINIT may be specified.

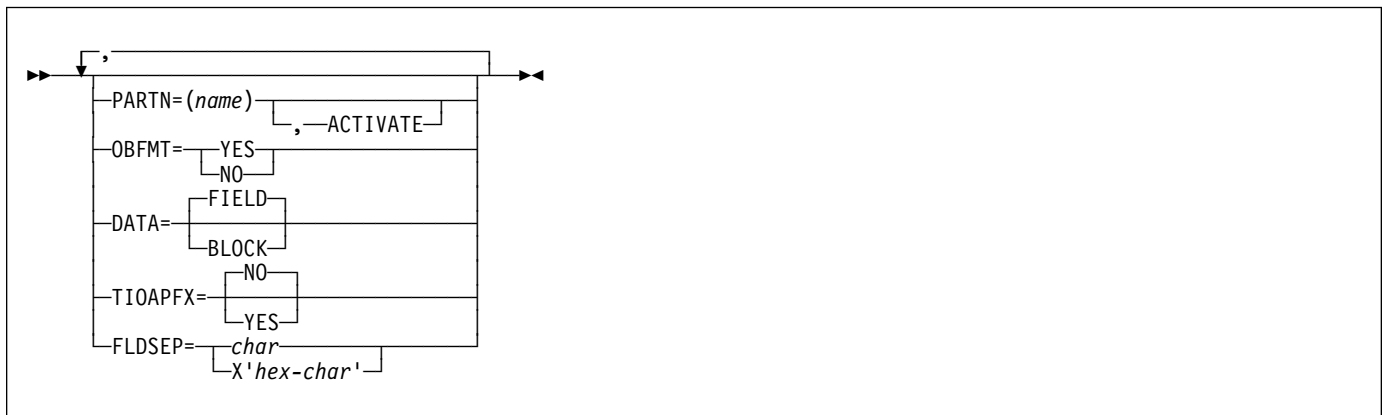
DFHMDI

Map definition.

Minimum BMS:

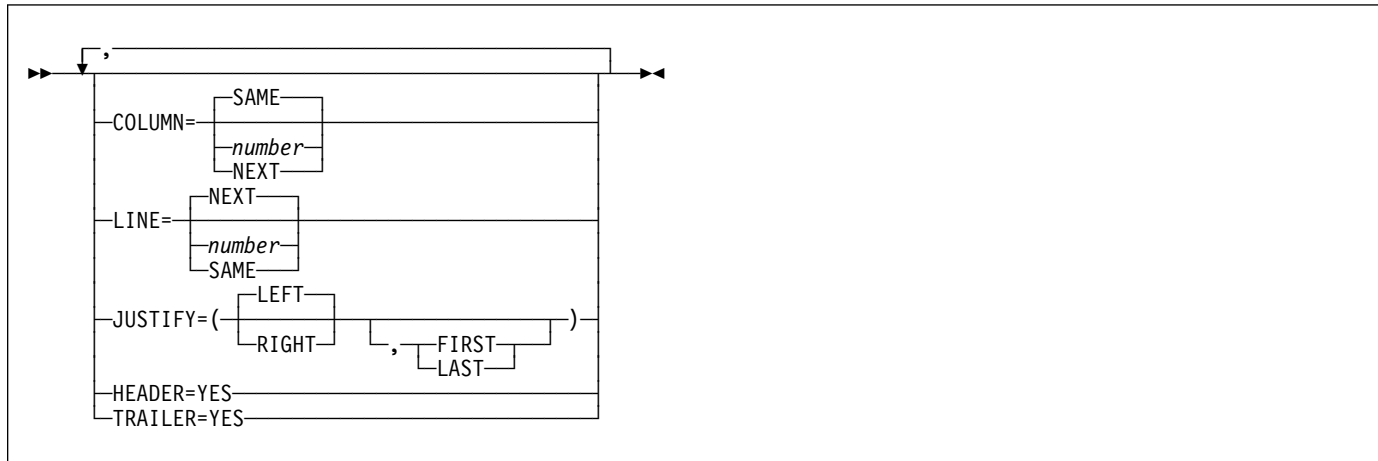


Standard BMS:



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 a physical map using an RDO DEFINE MAPSET command.

Operands

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.

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 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 411).

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 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 (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 411).

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)

MAPONLY is equivalent to:
MAPATTS=(COLOR,HILIGHT,PS,VALIDN)

DFHMDI

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.

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, 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.

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 DFHMSD 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 DFHMSD 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 DFHMSD 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.

TERM

kept for compatibility with previous releases.

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 DFHMSD 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.

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

DFHMDI

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.

VALIDN

specifies that:

- validation is to be used on an 8775 terminal
- this field can be processed by the BMS global user exits

This is overridden by the VALIDN operand of the DFHMDF macro, and overrides the VALIDN operand of the DFHMSD 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*.

USEREXIT

specifies that this field is to be processed by the BMS global user exits, XBMIN and XBMOU, if this field is received or transmitted in a 3270 datastream when the respective exit is enabled. For further information on the use of the BMS global user exits refer to the *CICS Customization Guide*.

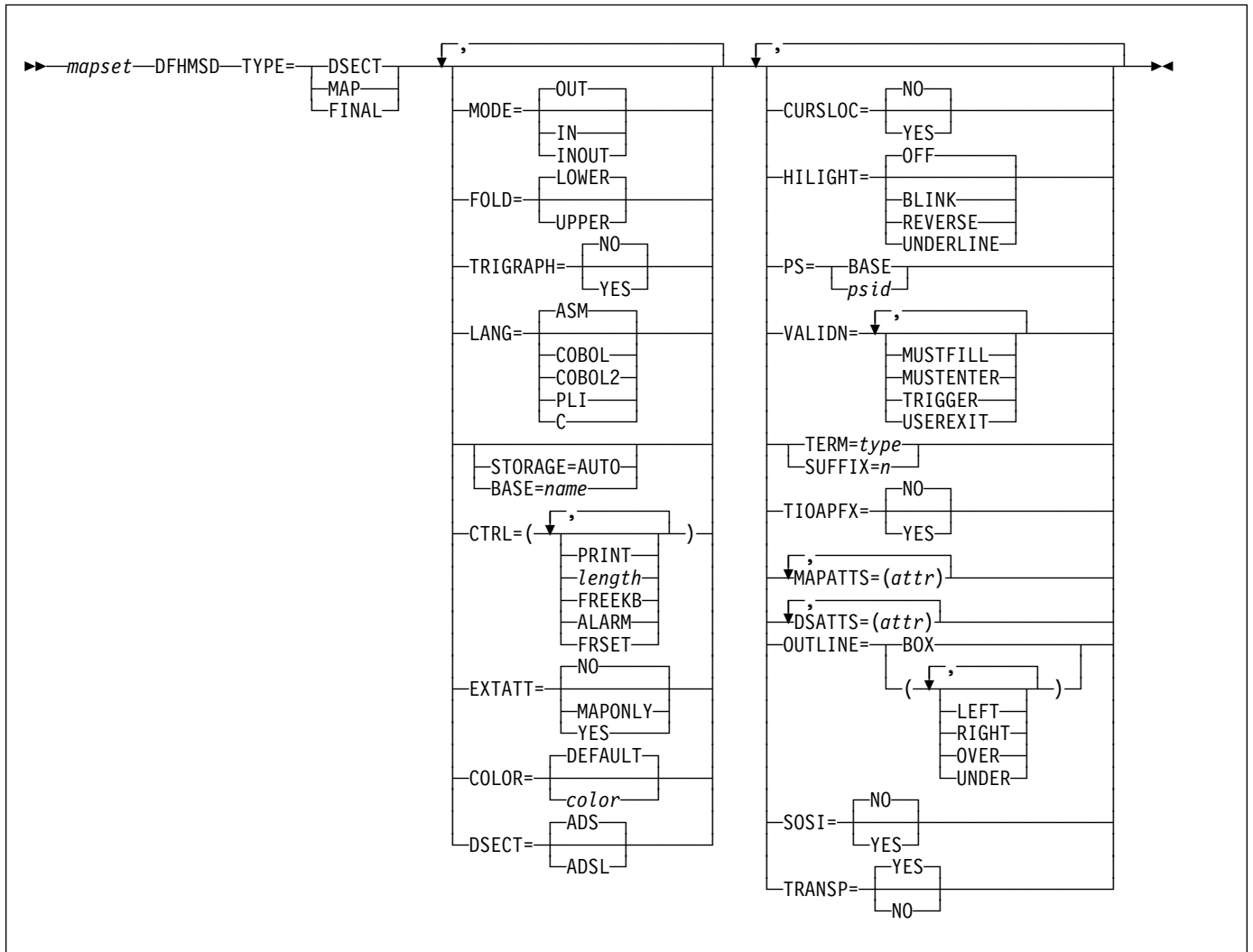
The MUSTFILL, MUSTENTER and TRIGGER specifications are valid only for terminals that support the field validation extended attribute, otherwise they are ignored. The USEREXIT specification applies to all 3270 devices.

Note: The USEREXIT specification is totally unconnected with the field validation extended attribute as defined in the 3270 datastream architecture.

DFHMSD

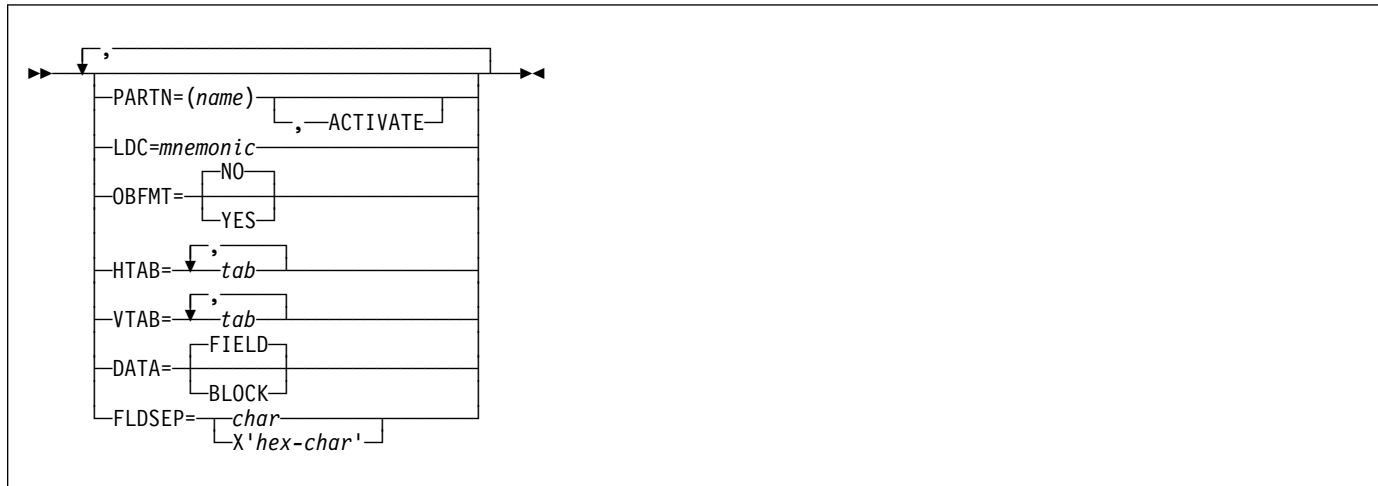
Mapset definition.

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 a physical map using an RDO DEFINE MAPSET command.

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.

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 411).

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 DFHMDF 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 411).

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.

DSECT

specifies whether the copybook generated supports the normal or long form of the application data structure (ADS).

ADS (default) requests mapping of the normal form of the ADS.

ADSL requests mapping of the long form of the ADS, where all fields are aligned on 4-byte boundaries. This form is required by the 3270 Bridge interface to MQSeries. See the *CICS External Interfaces Guide* for more information about the 3270 Bridge.

This option requires LANG=C.

EXTATT

this operand is supported for compatibility with previous releases. Each of the extended attributes can be defined

DFHMSD

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 TCAM and 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 27.

Table 27. 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:

DFHMSD

For TCAM-connected terminals (other than 3270 or SNA devices), use either CRLP or ALL; for TCAM-connected 3270s or SNA devices, select the appropriate parameter in the normal way.

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 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 Resource Definition Guide*.

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 C/370 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 Transaction Server for OS/390 Installation Guide*.

VALIDN

specifies that:

- validation is to be used on an 8775 terminal
- this field can be processed by the BMS global user exits

This is overridden by the VALIDN operand of the DFHMDI macro, which is in turn overridden by the VALIDN 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.

MUSTERER

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*.

USEREXIT

specifies that this field is to be processed by the BMS global user exits, XBMIN and XBMOU, if this field is received or transmitted in a 3270 datastream when the respective exit is enabled. For further information on the use of the BMS global user exits refer to the *CICS Customization Guide*.

| The MUSTFILL, MUSTENTER and TRIGGER
| specifications are valid only for terminals that support the
| field validation extended attribute, otherwise they are
| ignored. The USEREXIT specification applies to all 3270
| devices.

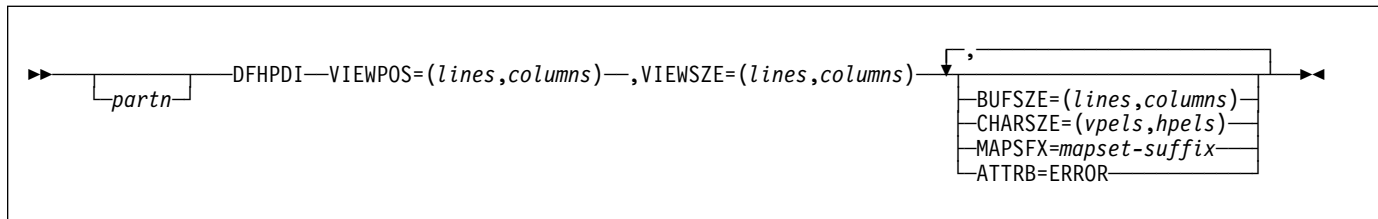
| **Note:** The USEREXIT specification is totally
| unconnected with the field validation extended
| attribute as defined in the 3270 datastream
| architecture.

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 ERRHIGHLIGHT is honored, but the LASTLINE option is ignored.

BUFSZE(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.

CHARSZE(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 CHARSZE 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 option ALTSUFFIX. If this operand is omitted, a suffix L is assumed if the “columns” value of the BUFSZE 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.

Index

Numerics

- 16MB line 140
- 2260 Display Station
 - CONVERSE command 55
 - RECEIVE command 232
 - SEND command 277
- 2265 Display Station
 - CONVERSE command 55
 - RECEIVE command 232
 - SEND command 277
- 2741 Communication Terminal
 - CONVERSE command 56
 - RECEIVE command 232
 - SEND command 277
- 2770 Data Communication System
 - CONVERSE command 54, 57
 - RECEIVE command 231
 - SEND command 276
- 2780 Data Transmission Terminal
 - CONVERSE command 54, 57
 - RECEIVE command 231
 - SEND command 276
- 2980 General Banking Terminal System
 - DFH2980 structure 234
 - output control 233
 - output to common buffer 234
 - passbook control 233
 - RECEIVE/SEND commands 233, 278
- 3270 Information Display System
 - (BTAM or TCAM supported) 58, 235, 278
 - (BTAM supported) 151, 171
 - logical unit 44, 152, 223, 265
- 3600 Finance Communication System
 - 3601 logical unit 44, 224, 266
 - 3614 logical unit 45, 225, 266
 - CONVERSE command 58
 - pipeline logical units 265
 - RECEIVE command 235
 - SEND command 279
- 3630 Plant Communication System
 - RECEIVE command 224
 - SEND command 266
- 3650 Host Command Processor
 - CONVERSE command 47
- 3650 Logical Units
 - RECEIVE command 225
- 3650 Store System
 - host conversational
 - LU 3270 267
 - LU 3653 268
 - interpreter logical unit 45, 157, 162, 225, 267
- 3650/3680 Full-function Logical Unit
 - RECEIVE command 227
 - SEND command 270
- 3650/3680 Store System
 - host command processor LU 268
- 3660 Supermarket Scanning System
 - CONVERSE command 54
 - RECEIVE command 231
 - SEND command 276
- 3680 Host Command Processor
 - CONVERSE command 47
- 3680 Programmable Store System
 - host command processor LU 268
- 3735 Programmable Buffered Terminal 60, 236, 279
- 3740 Data Entry System 60, 156, 157, 236
- 3767 Communication Terminal
 - interactive logical unit 47, 226, 269
- 3770 Data Communication System
 - batch logical unit 48, 226, 269
- 3770 Full Function Logical Unit
 - RECEIVE command 227
 - SEND command 270
- 3770 interactive logical unit
 - RECEIVE command 226
 - SEND command 269
- 3780 Data Communication Terminal
 - CONVERSE command 54
 - RECEIVE command 231
 - SEND command 276
- 3790 Communication System
 - 3270-display logical unit 49, 237, 271
 - Full Function Logical Unit 48
 - full-function logical unit 227, 270
 - SCS printer logical unit 270
- 4700 Finance Communication System
 - CONVERSE command 58
 - RECEIVE command 235
 - SEND command 279

A

- ABCODE option
 - ABEND command 13
 - ASSIGN command 23
- ABDUMP option
 - ASSIGN command 23
- ABEND AICA timeout 6
- ABEND command 13
- ABEND exit, reactivating 141
- abend support commands 9
- abnormal termination, task 141
- ABPROGRAM option
 - ASSIGN command 23

- absolute expressions 4
- ABSTIME option
 - ASKTIME command 21
 - FORMATTIME command 110
- access to system information
 - ADDRESS command 14
 - ADDRESS SET command 15
 - ASSIGN command 22
 - CICS storage areas 14, 15
- ACCUM option
 - SEND CONTROL command 283
 - SEND MAP command 287
 - SEND TEXT command 296
- ACEE option
 - ADDRESS command 14
- ACTION option
 - WRITE OPERATOR command 367
- ACTIVITY option
 - ASSIGN command 23
- ACTIVITYID option
 - ASSIGN command 23
- ACTPARTN option
 - SEND CONTROL command 283
 - SEND MAP command 287
 - SEND TEXT command 296
- ADDRESS command 14
- ADDRESS SET command 15
- address, cursor 397
- ADS descriptor 417
- ADS value
 - DFHMSD 435
- ADSL value
 - DFHMSD 435
- AFTER option
 - POST command 189
 - ROUTE command 259
 - START command 321
- AID option
 - RECEIVE MAP MAPPINGDEV command 242
- ALARM option
 - SEND CONTROL command 283
 - SEND MAP command 287
 - SEND MAP MAPPINGDEV command 291
 - SEND TEXT command 296
 - SEND TEXT NOEDIT command 302
- ALARM value
 - DFHMDI 429
 - DFHMSD 435
- ALIGNED attribute
 - PL/I 4
- ALL option
 - SEND PAGE command 293
- ALLOCATE (APPC) command 16
- ALLOCATE (LUTYPE6.1) command 18
- ALLOCATE (MRO) command 20
- ALLOCERR condition
 - SPOOLCLOSE command 308
 - SPOOLOPEN INPUT command 310
 - SPOOLOPEN OUTPUT command 313
 - SPOOLREAD command 315
 - SPOOLWRITE command 317
- ALTER option
 - QUERY SECURITY command 195
- ALTERNATE option
 - CONVERSE (non-VTAM) command 61
 - CONVERSE (VTAM) command 50
 - SEND (non-VTAM) command 280
 - SEND (VTAM) command 272
 - SEND CONTROL command 284
 - SEND MAP command 287
 - SEND TEXT command 297
 - SEND TEXT NOEDIT command 302
- ALTSCRN operand
 - DFHPSD 441
- ALTSCRNHT option
 - ASSIGN command 23
- ALTSCRNWD option
 - ASSIGN command 23
- ANYKEY option
 - HANDLE AID command 142
- APLKYBD option
 - ASSIGN command 23
- APLTEXT option
 - ASSIGN command 23
- APPC basic conversations
 - commands 9
- APPC logical unit
 - acquiring session to 16
 - initiating conversation with 39
 - returning mapped sessions to CICS 113
 - sending and receiving 41
- APPC mapped conversations
 - abending 145
 - commands 9
 - ensuring transmission of accumulated data 337
 - extracting attributes of 100
 - informing partner of error 161
 - issuing a positive response 150
 - receiving data 221
 - requesting change of direction 174
 - retrieving values from attach header 105
 - returning sessions to CICS 113
 - sending data 262
- application performance, monitoring 186
- APPLID option
 - ASSIGN command 23
- argument values
 - assembler language 4
 - C 2
 - COBOL 2
 - PL/I 3

- ASA option
 - SPOOLOPEN OUTPUT command 312
- ASIS option
 - CONVERSE (non-VTAM) command 61
 - CONVERSE (VTAM) command 50
 - RECEIVE (non-VTAM) command 238
 - RECEIVE (VTAM) command 228
 - RECEIVE MAP command 240
 - RECEIVE PARTN command 244
 - SEND (non-VTAM) command 280
- ASKIP value
 - DFHMDf 421
- ASKTIME command 21
- ASRAINTRPT option
 - ASSIGN command 23
- ASRAKEY option
 - ASSIGN command 23
- ASRAPSW option
 - ASSIGN command 24
- ASRAREGS option
 - ASSIGN command 24
- ASRASPC option
 - ASSIGN command 24
- ASRASTG option
 - ASSIGN command 24
- assembler language
 - argument values 4
 - LENGTH option default 4
 - program exit 390
 - register contents 390
 - translated code 390
- ASSIGN command 22
- asynchronous interrupt 394
- AT option
 - DOCUMENT INSERT command 82
 - POST command 190
 - ROUTE command 259
 - START command 321
- attach
 - start a task 324
- ATTACHID option
 - BUILD ATTACH (LUTYPE6.1) command 31
 - BUILD ATTACH (MRO) command 33
 - CONVERSE (non-VTAM) command 61
 - CONVERSE (VTAM) command 50
 - EXTRACT ATTACH (LUTYPE6.1) command 96
 - EXTRACT ATTACH (MRO) command 98
 - SEND (non-VTAM) command 280
 - SEND (VTAM) command 272
- attention identifier (AID) 142
- ATTRB operand
 - DFHMDf 421
 - DFHPDI 440
- attributes
 - control character list, DFHBMSCA 411

- AUTHENTICATE option
 - EXTRACT TCPIP command 107
- authentication commands 9
- AUTOPAGE option
 - SEND PAGE command 293
- AUXILIARY option
 - WRITEQ TS command 371

B

- back out to a syncpoint 331
- BASE operand
 - DFHMSD 434
- BASE value
 - DFHMDf 425
 - DFHMdI 431
 - DFHMSD 437
- basic mapping support (BMS)
 - ADS descriptor 417
 - commands 10
 - completing a logical message 293
 - deleting a logical message 191
 - determining input partition 244
 - field definition macro 417, 421
 - full BMS
 - RECEIVE MAP command 240
 - RECEIVE PARTN command 244
 - SEND CONTROL command 283
 - SEND MAP command 286
 - SEND PAGE 293
 - SEND PARTNSET 295
 - SEND TEXT command 296
 - SEND TEXT MAPPED 300
 - SEND TEXT NOEDIT 302
 - full-function BMS
 - PURGE MESSAGE 191
 - map definition macro 417, 428
 - mapping input data 240
 - mapping input data with MAPPINGDEV 242
 - mapset definition macro 417, 434
 - minimum BMS
 - RECEIVE MAP command 240
 - RECEIVE MAP MAPPINGDEV command 242
 - SEND CONTROL command 283
 - SEND MAP command 286
 - SEND MAP MAPPINGDEV command 291
 - partition definition macro 418, 440
 - partition set definition macro 418, 441
 - related constants 411
 - routing a logical message 259
 - sending previously mapped data 300
 - sending user-defined data stream 302
 - standard BMS
 - RECEIVE MAP command 240
 - RECEIVE PARTN command 244
 - SEND CONTROL command 283
 - SEND MAP command 286

- basic mapping support (BMS) *(continued)*
 - standard BMS *(continued)*
 - SEND PARTNSET 295
 - SEND TEXT command 296
- batch data interchange (BDI)
 - add record to data set 148
 - commands 10
 - conditions 163
 - delete a record from data set 158
 - read record from data set 168
 - request next record number 163
 - send data to output device 172
 - terminate data set 146, 155
 - update a record in data set 170
 - wait for function completion 176
- batch logical unit, 3770 48, 226, 269
- batch mode application, 3740 60, 236, 280
- BELOW option
 - GETMAIN command 140
- BIF DEEDIT command 30
- BINARY option
 - DOCUMENT INSERT command 82
- BLANK value
 - DFHMDF 423
- BLINK value
 - DFHMDF 423
 - DFHMDI 430
 - DFHMSD 436
- BLOCK value
 - DFHMDI 429
 - DFHMSD 435
- BOOKMARK option
 - DOCUMENT INSERT command 82
- BOTTOM value
 - DFHMDI 430
- BRDATA option
 - START BREXIT command 325
- BRDATALENGTH option
 - START BREXIT command 325
- BREXIT option
 - START BREXIT command 325
- bridge (3270)
 - start a task 325
- BRIDGE option
 - ASSIGN command 24
- browse operation
 - ending 90
 - read next record 204
 - read previous record 211
 - reset starting point 247
 - starting 327
- BRT value
 - DFHMDF 421
- BTAM programmable terminals 394
- BTRANS option
 - ASSIGN command 24

- BUFFER option
 - GDS RECEIVE command 131
 - RECEIVE (non-VTAM) command 238
 - RECEIVE (VTAM) command 228
- BUFSZE operand
 - DFHPDI 440
- BUILD ATTACH (LUTYPE6.1) command 31
- BUILD ATTACH (MRO) command 33
- built-in functions
 - commands 10

C

- C language
 - ADDRESS COMMAREA 14
 - ADDRESS EIB 14
 - argument values 2
 - LENGTH option default 3
 - translated code 389
- CADDRLENGTH option
 - EXTRACT TCPIP command 107
- CANCEL command 35
- CANCEL option
 - ABEND command 13
 - HANDLE ABEND command 142
- CARD option
 - ISSUE ABORT command 146
 - ISSUE END command 155
 - ISSUE SEND command 172
 - ISSUE WAIT command 176
- CASE operand
 - DFHMDF 422
- CBIDERR condition
 - ALLOCATE (APPC) command 17
 - ALLOCATE (LUTYPE6.1) command 18
 - CONVERSE (non-VTAM) command 62
 - CONVERSE (VTAM) command 52
 - EXTRACT ATTACH (LUTYPE6.1) command 97
 - EXTRACT ATTACH (MRO) command 99
 - SEND (non-VTAM) command 281
 - SEND (VTAM) command 273
- CBUFF option
 - SEND (non-VTAM) command 280
- CERTIFICATE option
 - EXTRACT CERTIFICATE command 102
- CHANGE PASSWORD command 37
- CHANGE TASK command 38
- CHANGED condition
 - DELETE command 69
 - REWRITE command 257
- CHANGETIME option
 - VERIFY PASSWORD command 336
- CHARSZ operand
 - DFHPDI 440
 - DFHPSD 441

CICS Web Interface (CWI) commands
 DOCUMENT CREATE 79
 DOCUMENT INSERT 82
 DOCUMENT RETRIEVE 84
 DOCUMENT SET 85
 EXTRACT CERTIFICATE 102
 EXTRACT WEB 348
 WEB ENDBROWSE FORMFIELD 346
 WEB ENDBROWSE HTTPHEADER 347
 WEB EXTRACT 348
 WEB READ FORMFIELD 349
 WEB READ HTTPHEADER 350
 WEB READNEXT FORMFIELD 351
 WEB READNEXT HTTPHEADER 352
 WEB RECEIVE 353
 WEB RETRIEVE 354
 WEB SEND 355
 WEB STARTBROWSE FORMFIELD 356
 WEB STARTBROWSE HTTPHEADER 357
 WEB WRITE 358
 CICS DATAKEY option
 GETMAIN command 140
 CLASS option
 SPOOL OPEN INPUT command 310
 SPOOL OPEN OUTPUT command 312
 CLEAR option
 HANDLE AID command 142
 client requests
 extracting information 102, 348
 CLIENTADDR option
 EXTRACT TCPIP command 107
 CLIENTADDRNU option
 EXTRACT TCPIP command 107
 CLIENTNAME option
 EXTRACT TCPIP command 107
 CLNTCODEPAGE option
 DOCUMENT RETRIEVE command 84
 WEB READ FORMFIELD command 349
 WEB RECEIVE command 353
 WEB SEND command 355
 WEB STARTBROWSE FORMFIELD command 356
 CLRPARTN option
 HANDLE AID command 142
 CMDSEC option
 ASSIGN command 24
 CNAMELENGTH option
 EXTRACT TCPIP command 107
 CNOTCOMPL option
 SEND (non-VTAM) command 280
 SEND (VTAM) command 272
 COBOL
 argument values 2
 translated code 389
 CODEREG operand 391
 COLOR operand
 DFHMDF 422
 COLOR operand (*continued*)
 DFHMDF 428
 DFHMDS 434
 COLOR option
 ASSIGN command 24
 COLUMN operand
 DFHMDF 428
 column value
 DFHMDF 431
 command language translator
 translated code 390
 commands
 format, arguments 1
 scheduling 11
 security 11
 spool 11
 TCP/IP 11
 temporary storage control 11
 COMMAREA option
 ADDRESS command 14
 LINK command 179
 RETURN command 252
 XCTL command 374
 common buffer, output to, 2980 234
 common programming interface communications (CPI
 communications) 401
 COMMONNAME option
 EXTRACT CERTIFICATE command 102
 COMMONNAMLEN option
 EXTRACT CERTIFICATE command 102
 COMPAREMAX option
 GET COUNTER command 136
 GET DCOUNTER command 136
 UPDATE COUNTER command 334
 UPDATE DCOUNTER command 334
 COMPAREMIN option
 GET COUNTER command 136
 GET DCOUNTER command 136
 UPDATE COUNTER command 334
 UPDATE DCOUNTER command 334
 COMPLETE option
 DUMP TRANSACTION command 87
 CONFIRM option
 GDS SEND command 133
 SEND (VTAM) command 272
 CONNECT PROCESS command 39
 CONSISTENT option
 READ command 198
 READNEXT command 205
 READPREV command 212
 CONSOLE option
 ISSUE ABORT command 146
 ISSUE END command 155
 ISSUE SEND command 172
 ISSUE WAIT command 176

console support commands 10

constants

- AID values, DFHAID 415
- attribute values, DFHBMSCA 411
- for 3270 attributes 411
- for examining EIBAID field 415
- for MSR control values 413
- for printer format controls 411
- MSR control, DFHtex read 413
- printer control values, DFHBMSCA 411

CONTROL option

- QUERY SECURITY command 195

CONVDATA option

- GDS CONNECT PROCESS command 121
- GDS EXTRACT ATTRIBUTES command 123
- GDS FREE command 125
- GDS ISSUE ABEND command 126
- GDS ISSUE CONFIRMATION command 127
- GDS ISSUE ERROR command 128
- GDS ISSUE PREPARE command 129
- GDS ISSUE SIGNAL command 130
- GDS RECEIVE command 131
- GDS SEND command 133
- GDS WAIT command 135

CONVERSE (2260) command 55

CONVERSE (2741) command 56

CONVERSE (2770) command 57

CONVERSE (2780) command 57

CONVERSE (3270 display) command 58

CONVERSE (3270 logical) command 44

CONVERSE (3600 BTAM) command 58

CONVERSE (3600-3601) command 44

CONVERSE (3600-3614) command 45

CONVERSE (3650 interpreter) command 45

CONVERSE (3650-3270) command 46

CONVERSE (3650-3653) command 46

CONVERSE (3650-3680) command 47

CONVERSE (3735) command 60

CONVERSE (3740) command 60

CONVERSE (3767) command 47

CONVERSE (3770) command 48

CONVERSE (3790 3270-display) command 49

CONVERSE (3790 full-function or inquiry) command 48

CONVERSE (APPC) command 41

CONVERSE (LUTYPE2/LUTYPE3) command 42

CONVERSE (LUTYPE4) command 42

CONVERSE (LUTYPE6.1) command 43

CONVERSE (MRO) command 53

CONVERSE (non-VTAM default) command 53

CONVERSE (SCS) command 43

CONVERSE (System/3) command 54

CONVERSE (System/7) command 54

CONVERSE (VTAM default) command 41

CONVERSE option

- ISSUE LOAD command 162

converse with terminal or LU 394

CONVID option

- CONNECT PROCESS command 39
- CONVERSE (non-VTAM) command 61
- CONVERSE (VTAM) command 50
- EXTRACT ATTACH (LUTYPE6.1) command 96
- EXTRACT ATTACH (MRO) command 98
- EXTRACT ATTRIBUTES (APPC) command 100
- EXTRACT ATTRIBUTES (MRO) command 101
- EXTRACT PROCESS command 105
- FREE (APPC) command 113
- FREE (LUTYPE6.1) command 114
- FREE (MRO) command 115
- GDS ALLOCATE command 118
- GDS CONNECT PROCESS command 121
- GDS EXTRACT ATTRIBUTES command 123
- GDS EXTRACT PROCESS command 124
- GDS FREE command 125
- GDS ISSUE ABEND command 126
- GDS ISSUE CONFIRMATION command 127
- GDS ISSUE ERROR command 128
- GDS ISSUE PREPARE command 129
- GDS ISSUE SIGNAL command 130
- GDS RECEIVE command 131
- GDS SEND command 133
- GDS WAIT command 135
- ISSUE ABEND command 145
- ISSUE CONFIRMATION command 150
- ISSUE ERROR command 161
- ISSUE PREPARE command 165
- ISSUE SIGNAL (APPC) command 174
- ISSUE SIGNAL (LUTYPE6.1) command 175
- POINT command 188
- RECEIVE (VTAM) command 228
- SEND (VTAM) command 272
- WAIT CONVID command 337
- WAIT TERMINAL command 344

copy displayed information 396

copybooks

- DFHAID 415
- DFHBMSCA 411
- DFHEIBLK 391
- DFHMSRCA 413

COUNTER option

- DEFINE COUNTER command 64
- DEFINE DCOUNTER command 64
- DELETE COUNTER command 73
- GET COUNTER command 136
- QUERY COUNTER command 193
- REWIND COUNTER command 254
- UPDATE COUNTER command 334

COUNTRY option

- EXTRACT CERTIFICATE command 102

COUNTRYLEN option

- EXTRACT CERTIFICATE command 102

- CPI communications (SAA) 401
- create a journal record 177, 364
- CSMT transaction
 - resend message destination 59
- CSTL transaction
 - resend message destination 59
- CTLCHAR option
 - CONVERSE (non-VTAM) command 61
 - CONVERSE (VTAM) command 50
 - ISSUE COPY (3270 display) command 151
 - ISSUE COPY (3270 logical) command 152
 - SEND (non-VTAM) command 280
 - SEND (VTAM) command 272
- CTRL operand
 - DFHMDI 428
 - DFHMSD 434
- CURRENT option
 - SEND PAGE command 293
- CURSLOC operand
 - DFHMDI 429
 - DFHMSD 435
- cursor address 397
- CURSOR option
 - RECEIVE MAP MAPPINGDEV command 242
 - SEND CONTROL command 284
 - SEND MAP command 287
 - SEND MAP MAPPINGDEV command 291
 - SEND TEXT command 297
- cursor position
 - terminal control 397
- CVDA (CICS-value data area)
 - argument values 2
 - command format 2
 - passing and receiving 5
 - values in alphabetic order 407
 - values in numeric order 408
- CVDA options
 - ACTION
 - WRITE OPERATOR command 367
 - ALTER
 - QUERY SECURITY command 195
 - ASRAKEY
 - ASSIGN command 23
 - ASRASPC
 - ASSIGN command 24
 - CONTROL
 - QUERY SECURITY command 195
 - LOGMESSAGE
 - QUERY SECURITY command 195
 - MAXLIFETIME
 - DEQ 77
 - ENQ 92
 - PURGEABILITY
 - WAIT EXTERNAL 340
 - WAITCICS 345
 - READ
 - QUERY SECURITY command 195

- CVDA options (*continued*)
 - STATE 51, 62, 229, 239, 273, 281
 - ALLOCATE (APPC) 16
 - ALLOCATE (MRO) 20
 - CONNECT PROCESS 39
 - EXTRACT ATTRIBUTES (APPC) 100
 - EXTRACT ATTRIBUTES (MRO) 101
 - FREE (APPC) 113
 - FREE (MRO) 115
 - GDS ALLOCATE 118
 - GDS CONNECT PROCESS 121
 - GDS EXTRACT ATTRIBUTES 123
 - GDS FREE 125
 - GDS ISSUE ABEND 126
 - GDS ISSUE CONFIRMATION 127
 - GDS ISSUE ERROR 128
 - GDS ISSUE PREPARE 129
 - GDS ISSUE SIGNAL 130
 - GDS RECEIVE 132
 - GDS SEND 133
 - GDS WAIT 135
 - ISSUE ABEND command 145
 - ISSUE CONFIRMATION 150
 - ISSUE ERROR 161
 - ISSUE PREPARE 165
 - ISSUE SIGNAL (APPC) 174
 - WAIT CONVID 337
 - UPDATE
 - QUERY SECURITY command 196
- CVDA values
 - ALLOCATED
 - ALLOCATE (APPC) 16
 - ALLOCATE (MRO) 20
 - CONNECT PROCESS 39
 - EXTRACT ATTRIBUTES (APPC) 100
 - EXTRACT ATTRIBUTES (MRO) 101
 - FREE (APPC) 113
 - FREE (MRO) 115
 - GDS ALLOCATE 118
 - GDS CONNECT PROCESS 121
 - GDS EXTRACT ATTRIBUTES 123
 - GDS FREE 125
 - GDS ISSUE ABEND 126
 - GDS ISSUE CONFIRMATION 127
 - GDS ISSUE ERROR 128
 - GDS ISSUE PREPARE 129
 - GDS ISSUE SIGNAL 130
 - GDS RECEIVE 132
 - GDS SEND 133
 - GDS WAIT 135
 - ISSUE ABEND command 145
 - ISSUE CONFIRMATION 150
 - ISSUE ERROR 161
 - ISSUE PREPARE 165
 - ISSUE SIGNAL (APPC) 174
 - RECEIVE (APPC) command 229
 - RECEIVE (MRO) command 239

CVDA values (continued)

ALLOCATED (continued)

SEND (APPC) command 273
SEND (MRO) command 281
WAIT CONVID 337

ALTERABLE

QUERY SECURITY command 195

BASESPACE

ASSIGN command 24

CICSEXECKEY

ASSIGN command 23

CONFFREE

CONNECT PROCESS 39
EXTRACT ATTRIBUTES (APPC) 100
FREE (APPC) 113
GDS ALLOCATE 118
GDS CONNECT PROCESS 121
GDS EXTRACT ATTRIBUTES 123
GDS FREE 125
GDS ISSUE ABEND 126
GDS ISSUE CONFIRMATION 127
GDS ISSUE ERROR 128
GDS ISSUE PREPARE 129
GDS ISSUE SIGNAL 130
GDS RECEIVE 132
GDS SEND 133
GDS WAIT 135
ISSUE ABEND command 145
ISSUE CONFIRMATION 150
ISSUE ERROR 161
ISSUE PREPARE 165
ISSUE SIGNAL (APPC) 174
RECEIVE (APPC) command 229
SEND (APPC) command 273
WAIT CONVID 337

CONFRECEIVE

CONNECT PROCESS 39
EXTRACT ATTRIBUTES (APPC) 100
FREE (APPC) 113
GDS ALLOCATE 118
GDS CONNECT PROCESS 121
GDS EXTRACT ATTRIBUTES 123
GDS FREE 125
GDS ISSUE ABEND 126
GDS ISSUE CONFIRMATION 127
GDS ISSUE ERROR 128
GDS ISSUE PREPARE 129
GDS ISSUE SIGNAL 130
GDS RECEIVE 132
GDS SEND 133
GDS WAIT 135
ISSUE ABEND command 145
ISSUE CONFIRMATION 150
ISSUE ERROR 161
ISSUE PREPARE 165
ISSUE SIGNAL (APPC) 174
RECEIVE (APPC) command 229

CVDA values (continued)

CONFRECEIVE (continued)

SEND (APPC) command 273
WAIT CONVID 337

CONFSEND

CONNECT PROCESS 39
EXTRACT ATTRIBUTES (APPC) 100
FREE (APPC) 113
GDS ALLOCATE 118
GDS CONNECT PROCESS 121
GDS EXTRACT ATTRIBUTES 123
GDS FREE 125
GDS ISSUE ABEND 126
GDS ISSUE CONFIRMATION 127
GDS ISSUE ERROR 128
GDS ISSUE PREPARE 129
GDS ISSUE SIGNAL 130
GDS RECEIVE 132
GDS SEND 133
GDS WAIT 135
ISSUE ABEND command 145
ISSUE CONFIRMATION 150
ISSUE ERROR 161
ISSUE PREPARE 165
ISSUE SIGNAL (APPC) 174
RECEIVE (APPC) command 229
SEND (APPC) command 273
WAIT CONVID 337

CRITICAL

WRITE OPERATOR command 367

CTRLABLE

QUERY SECURITY command 195

EVENTUAL

WRITE OPERATOR command 367

FREE

CONNECT PROCESS 39
EXTRACT ATTRIBUTES (APPC) 100
EXTRACT ATTRIBUTES (MRO) 101
FREE (APPC) 113
FREE (MRO) 115
GDS ALLOCATE 118
GDS CONNECT PROCESS 121
GDS EXTRACT ATTRIBUTES 123
GDS FREE 125
GDS ISSUE ABEND 126
GDS ISSUE CONFIRMATION 127
GDS ISSUE ERROR 128
GDS ISSUE PREPARE 129
GDS ISSUE SIGNAL 130
GDS RECEIVE 132
GDS SEND 133
GDS WAIT 135
ISSUE ABEND command 145
ISSUE CONFIRMATION 150
ISSUE ERROR 161
ISSUE PREPARE 165
ISSUE SIGNAL (APPC) 174

CVDA values (continued)

FREE (continued)

RECEIVE (APPC) command 229
 RECEIVE (MRO) command 239
 SEND (APPC) command 273
 SEND (MRO) command 281
 WAIT CONVID 337

IMMEDIATE

WRITE OPERATOR command 367

LOG

QUERY SECURITY command 195

NOLOG

QUERY SECURITY command 195

NONCICS

ASSIGN command 23

NOTALTERABLE

QUERY SECURITY command 195

NOTAPPLIC

ASSIGN command 24

NOTCTRLABLE

QUERY SECURITY command 195

NOTPURGEABLE

WAIT EXTERNAL 340

WAITCICS 345

NOTREADABLE

QUERY SECURITY command 195

NOTUPDATABLE

QUERY SECURITY command 196

PENDFREE

CONNECT PROCESS 39
 EXTRACT ATTRIBUTES (APPC) 100
 EXTRACT ATTRIBUTES (MRO) 101
 FREE (APPC) 113
 FREE (MRO) 115
 GDS ALLOCATE 118
 GDS CONNECT PROCESS 121
 GDS EXTRACT ATTRIBUTES 123
 GDS FREE 125
 GDS ISSUE ABEND 126
 GDS ISSUE CONFIRMATION 127
 GDS ISSUE ERROR 128
 GDS ISSUE PREPARE 129
 GDS ISSUE SIGNAL 130
 GDS RECEIVE 132
 GDS SEND 133
 GDS WAIT 135
 ISSUE ABEND command 145
 ISSUE CONFIRMATION 150
 ISSUE ERROR 161
 ISSUE PREPARE 165
 ISSUE SIGNAL (APPC) 174
 RECEIVE (APPC) command 229
 RECEIVE (MRO) command 239
 SEND (APPC) command 273
 SEND (MRO) command 281
 WAIT CONVID 337

CVDA values (continued)

PENDRECEIVE

CONNECT PROCESS 39
 EXTRACT ATTRIBUTES (APPC) 100
 FREE (APPC) 113
 GDS ALLOCATE 118
 GDS CONNECT PROCESS 121
 GDS EXTRACT ATTRIBUTES 123
 GDS FREE 125
 GDS ISSUE ABEND 126
 GDS ISSUE CONFIRMATION 127
 GDS ISSUE ERROR 128
 GDS ISSUE PREPARE 129
 GDS ISSUE SIGNAL 130
 GDS RECEIVE 132
 GDS SEND 133
 GDS WAIT 135
 ISSUE ABEND command 145
 ISSUE CONFIRMATION 150
 ISSUE ERROR 161
 ISSUE PREPARE 165
 ISSUE SIGNAL (APPC) 174
 RECEIVE (APPC) command 229
 SEND (APPC) command 273
 WAIT CONVID 337

PURGEABLE

WAIT EXTERNAL 340

WAITCICS 345

READABLE

QUERY SECURITY command 195

RECEIVE

CONNECT PROCESS 39
 CONVERSE command (MRO) 62
 EXTRACT ATTRIBUTES (APPC) 100
 EXTRACT ATTRIBUTES (MRO) 101
 FREE (APPC) 113
 FREE (MRO) 115
 GDS ALLOCATE 118
 GDS CONNECT PROCESS 121
 GDS EXTRACT ATTRIBUTES 123
 GDS FREE 125
 GDS ISSUE ABEND 126
 GDS ISSUE CONFIRMATION 127
 GDS ISSUE ERROR 128
 GDS ISSUE PREPARE 129
 GDS ISSUE SIGNAL 130
 GDS RECEIVE 132
 GDS SEND 133
 GDS WAIT 135
 ISSUE ABEND command 145
 ISSUE CONFIRMATION 150
 ISSUE ERROR 161
 ISSUE PREPARE 165
 ISSUE SIGNAL (APPC) 174
 RECEIVE (APPC) command 229
 RECEIVE (MRO) command 239
 SEND (APPC) command 273

CVDA values (continued)

RECEIVE (continued)

SEND (MRO) command 281
WAIT CONVID 337

ROLLBACK

CONNECT PROCESS 39
CONVERSE command (MRO) 62
EXTRACT ATTRIBUTES (APPC) 100
EXTRACT ATTRIBUTES (MRO) 101
FREE (APPC) 113
FREE (MRO) 115
GDS ALLOCATE 118
GDS CONNECT PROCESS 121
GDS EXTRACT ATTRIBUTES 123
GDS FREE 125
GDS ISSUE ABEND 126
GDS ISSUE CONFIRMATION 127
GDS ISSUE ERROR 128
GDS ISSUE PREPARE 129
GDS ISSUE SIGNAL 130
GDS RECEIVE 132
GDS SEND 133
GDS WAIT 135
ISSUE ABEND command 145
ISSUE CONFIRMATION 150
ISSUE ERROR 161
ISSUE PREPARE 165
ISSUE SIGNAL (APPC) 174
RECEIVE (APPC) command 229
RECEIVE (MRO) command 239
SEND (APPC) command 273
SEND (MRO) command 281
WAIT CONVID 337

SEND

CONNECT PROCESS 39
CONVERSE command (MRO) 62
EXTRACT ATTRIBUTES (APPC) 100
EXTRACT ATTRIBUTES (MRO) 101
FREE (APPC) 113
FREE (MRO) 115
GDS ALLOCATE 118
GDS CONNECT PROCESS 121
GDS EXTRACT ATTRIBUTES 123
GDS FREE 125
GDS ISSUE ABEND 126
GDS ISSUE CONFIRMATION 127
GDS ISSUE ERROR 128
GDS ISSUE PREPARE 129
GDS ISSUE SIGNAL 130
GDS RECEIVE 132
GDS SEND 133
GDS WAIT 135
ISSUE ABEND command 145
ISSUE CONFIRMATION 150
ISSUE ERROR 161
ISSUE PREPARE 165
ISSUE SIGNAL (APPC) 174

CVDA values (continued)

SEND (continued)

RECEIVE (APPC) command 229
RECEIVE (MRO) command 239
SEND (APPC) command 273
SEND (MRO) command 281
WAIT CONVID 337

SUBSPACE

ASSIGN command 24

SYCNCFREE

CONNECT PROCESS 39
CONVERSE command (MRO) 62
EXTRACT ATTRIBUTES (APPC) 100
EXTRACT ATTRIBUTES (MRO) 101
FREE (APPC) 113
FREE (MRO) 115
GDS ALLOCATE 118
GDS CONNECT PROCESS 121
GDS EXTRACT ATTRIBUTES 123
GDS FREE 125
GDS ISSUE ABEND 126
GDS ISSUE CONFIRMATION 127
GDS ISSUE ERROR 128
GDS ISSUE PREPARE 129
GDS ISSUE SIGNAL 130
GDS RECEIVE 132
GDS SEND 133
GDS WAIT 135
ISSUE ABEND command 145
ISSUE CONFIRMATION 150
ISSUE ERROR 161
ISSUE PREPARE 165
ISSUE SIGNAL (APPC) 174
RECEIVE (APPC) command 229
RECEIVE (MRO) command 239
SEND (APPC) command 273
SEND (MRO) command 281
WAIT CONVID 337

SYNCRECEIVE

CONNECT PROCESS 39
CONVERSE command (MRO) 62
EXTRACT ATTRIBUTES (APPC) 100
EXTRACT ATTRIBUTES (MRO) 101
FREE (APPC) 113
FREE (MRO) 115
GDS ALLOCATE 118
GDS CONNECT PROCESS 121
GDS EXTRACT ATTRIBUTES 123
GDS FREE 125
GDS ISSUE ABEND 126
GDS ISSUE CONFIRMATION 127
GDS ISSUE ERROR 128
GDS ISSUE PREPARE 129
GDS ISSUE SIGNAL 130
GDS RECEIVE 132
GDS SEND 133
GDS WAIT 135

CVDA values (continued)

SYNCRECEIVE (continued)

ISSUE ABEND command 145
ISSUE CONFIRMATION 150
ISSUE ERROR 161
ISSUE PREPARE 165
ISSUE SIGNAL (APPC) 174
RECEIVE (APPC) command 229
RECEIVE (MRO) command 239
SEND (APPC) command 273
SEND (MRO) command 281
WAIT CONVID 337

SYNCSEND

CONNECT PROCESS 39
CONVERSE command (MRO) 62
EXTRACT ATTRIBUTES (APPC) 100
EXTRACT ATTRIBUTES (MRO) 101
FREE (APPC) 113
FREE (MRO) 115
GDS ALLOCATE 118
GDS CONNECT PROCESS 121
GDS EXTRACT ATTRIBUTES 123
GDS FREE 125
GDS ISSUE ABEND 126
GDS ISSUE CONFIRMATION 127
GDS ISSUE ERROR 128
GDS ISSUE PREPARE 129
GDS ISSUE SIGNAL 130
GDS RECEIVE 132
GDS SEND 133
GDS WAIT 135
ISSUE ABEND command 145
ISSUE CONFIRMATION 150
ISSUE ERROR 161
ISSUE PREPARE 165
ISSUE SIGNAL (APPC) 174
RECEIVE (APPC) command 229
RECEIVE (MRO) command 239
SEND (APPC) command 273
SEND (MRO) command 281
WAIT CONVID 337

TASK

DEQ 77
ENQ 93

UOW

DEQ 77
ENQ 93

UPDATABLE

QUERY SECURITY command 196

USEREXECKEY

ASSIGN command 23

CWA option

ADDRESS command 14

CWALENG option

ASSIGN command 24

D

data

passing to new tasks 321

DATA operand

DFHMDI 429
DFHMSD 435

DATA option

FREEMAIN command 116

data sets

add records to 148
interrogating 167
processing termination 155
read records from 168
update records 170

data tables

CICS/user-maintained/coupling facility

DELETE command 68
ENDBR command 90
READ command 197
READNEXT command 204
READPREV command 211
RESETBR command 247
REWRITE command 256
STARTBR command 327
UNLOCK command 332
WRITE command 359

data to output device, sending 172

data-area argument

CICS command format 2
definition 1

data-value argument

CICS command format 2
definition 1

data, deleting

file control records 68
named counter 73
temporary storage queues 76
transient data queues 75

DATA1 option

MONITOR command 186

DATA2 option

MONITOR command 186

DATALength option

LINK command 179

DATAONLY option

DOCUMENT RETRIEVE command 84
SEND MAP command 287
SEND MAP MAPPINGDEV command 291

DATAPOINTER option

FREEMAIN command 116

DATAREG operand 391

DATASTR option

BUILD ATTACH (LUTYPE6.1) command 31
BUILD ATTACH (MRO) command 33
EXTRACT ATTACH (LUTYPE6.1) command 96

DATASTR option (*continued*)
 EXTRACT ATTACH (MRO) command 98
 DATE option
 FORMATTIME command 110
 DATEFORM option
 FORMATTIME command 110
 DATESEP option
 FORMATTIME command 110
 DAYCOUNT option
 FORMATTIME command 110
 DAYOFMONTH option
 FORMATTIME command 111
 DAYOFWEEK option
 FORMATTIME command 111
 DAYSLEFT option
 VERIFY PASSWORD command 336
 DCOUNTER option
 DELETE DCOUNTER command 73
 GET DCOUNTER command 137
 QUERY DCOUNTER command 193
 REWIND DCOUNTER command 254
 UPDATE DCOUNTER command 334
 DCT option
 DUMP TRANSACTION command 87
 DDMMYY option
 FORMATTIME command 111
 DDMMYYYY option
 FORMATTIME command 111
 DEBKEY option
 READ command 198
 STARTBR command 327
 DEBREC option
 READ command 198
 STARTBR command 327
 DEFAULT option
 CONVERSE (non-VTAM) command 61
 CONVERSE (VTAM) command 50
 SEND (non-VTAM) command 280
 SEND (VTAM) command 272
 SEND CONTROL command 284
 SEND MAP command 287
 SEND TEXT command 297
 SEND TEXT NOEDIT command 302
 DEFINE COUNTER command 64
 DEFINE DCOUNTER command 64
 DEFRESP option
 CONVERSE (non-VTAM) command 61
 CONVERSE (VTAM) command 50
 ISSUE ADD command 148
 ISSUE ERASE command 158
 ISSUE REPLACE command 170
 ISSUE SEND command 172
 SEND (non-VTAM) command 280
 SEND (VTAM) command 272
 DEFSCRNHT option
 ASSIGN command 25
 DEFSCRNWD option
 ASSIGN command 25
 DELAY command 66
 delay processing, task 66
 DELETE command 68
 DELETE COUNTER command 73
 DELETE DCOUNTER command 73
 delete loaded program 245
 DELETE option
 SPOOLCLOSE command 308
 delete records
 batch data interchange records 158
 DELETEQ TD command 75
 DELETEQ TS command 76
 deleting data
 named counter 73
 temporary storage queues 76
 transient data queues 75
 DELIMITER option
 ASSIGN command 25
 DEQ command 77
 dequeue from resource 77
 DEST option
 CONVERSE (non-VTAM) command 61
 CONVERSE (VTAM) command 50
 SEND (non-VTAM) command 280
 SEND (VTAM) command 272
 DESTCOUNT option
 ASSIGN command 25
 DESTID option
 ASSIGN command 25
 ISSUE ABORT command 146
 ISSUE ADD command 148
 ISSUE END command 155
 ISSUE ERASE command 158
 ISSUE NOTE command 163
 ISSUE QUERY command 167
 ISSUE REPLACE command 170
 ISSUE SEND command 172
 ISSUE WAIT command 176
 DESTIDLENG option
 ASSIGN command 25
 ISSUE ABORT command 146
 ISSUE ADD command 148
 ISSUE END command 155
 ISSUE ERASE command 158
 ISSUE NOTE command 163
 ISSUE QUERY command 167
 ISSUE REPLACE command 170
 ISSUE SEND command 172
 ISSUE WAIT command 176
 DET value
 DFHMDF 421
 DFH2980 structure 234
 DFHAID attention identifier list 415

DFHBMSCA, standard attribute and printer control character list, BMS 411

DFHEAL interface processor 390

DFHECALL macro 390

DFHEIBLK copybook 391

DFHEICAL macro, use DFHECALL 390

DFHEIEND macro 390

DFHEIENT macro

- CODEREG 391
- DATAREG 391
- defaults 391
- description 390
- EIBREG 391

DFHEIGBL macro 390

DFHEIPLR symbolic register 392

DFHEIRET macro 390

DFHEISTG macro 391

DFHMDF macro 420

DFHMDI macro 427

DFHMIRS 180

DFHMSD macro 433

DFHMSRCA, MSR control value constants 413

DFHPDI macro 440

DFHPSD macro 441

DFHRESP, built-in function 5

DFHVALUE, translator routine 5

diagnostic services commands 10

DISABLED condition

- DELETE command 69
- DELETEQ TD command 75
- READ command 200
- READQ TD command 218
- STARTBR command 328
- UNLOCK command 332
- WRITE command 360
- WRITEQ TD command 369

disconnect a switched line 394

display-device operations

- attention identifier (AID) 397
- attention identifier list, DFHAID 415
- copy displayed information 396
- cursor address 397
- erase all unprotected fields 396
- input operation without data 397
- pass control on receipt of an AID 142, 144
- print displayed information 396
- standard attribute and printer control character list, DFHBMSCA 411
- terminal 396

distributed program link (DPL) 403

DOCSIZE option

- DOCUMENT INSERT command 82

DOCTOKEN option

- DOCUMENT RETRIEVE command 84
- DOCUMENT SET command 85
- WEB RETRIEVE command 354

DOCTOKEN option (*continued*)

- WEB SEND command 355

document

- adding symbols to symbol table 85
- creating 79
- sending documents 355

DOCUMENT CREATE command 79

DOCUMENT INSERT command 82

DOCUMENT option

- DOCUMENT INSERT command 82

DOCUMENT RETRIEVE command 84

document services

- commands 10

DOCUMENT SET command 85

DPL, distributed program link 403

DRK value

- DFHMDF 422

DS3270 option

- ASSIGN command 25

DSATTS operand

- DFHMDI 429
- DFHMSD 435

DSECT operand

- DFHMSD 435

DSECT value

- DFHMSD 438

DSSCS option

- ASSIGN command 25

DSSTAT condition

- ISSUE RECEIVE command 168

DUMP TRANSACTION command 87

DUMPCODE option

- DUMP TRANSACTION command 87

DUPKEY condition

- DELETE command 69
- READ command 200
- READNEXT command 208
- READPREV command 214

DUPREC condition

- REWRITE command 257
- WRITE command 360

dynamic allocation 312

dynamic storage, extensions 391

E

ECADDR option

- WAIT EVENT command 339

ECBLIST option

- WAIT EXTERNAL command 340
- WAITCICS command 345

EDF, execution diagnostic facility 323

EIB fields

- EIBAID 377
- EIBATT 377
- EIBCALEN 377

EIB fields (*continued*)

EIBCOMPL 377
EIBCONF 377
EIBCPOSN 377
EIBDATE 377
EIBDS 377
EIBEOC 378
EIBERR 378
EIBERRCD 378
EIBFMH 378
EIBFN 378
EIBFREE 380
EIBNODAT 380
EIBRCODE 380
EIBRECV 383
EIBREQID 383
EIBRESP 383
EIBRESP2 384
EIBRLDBK 384
EIBRSRCE 384
EIBSIG 384
EIBSYNC 385
EIBSYNRB 385
EIBTASKN 385
EIBTIME 385
EIBTRMID 385
EIBTRNID 385

EIB option
ADDRESS command 14

EIBAID 397
examining contents of field 415

EIBREG operand 391

ENDACTIVITY option
RETURN command 252

ENDBR command 90

ENDDATA condition
RETRIEVE command 250

ENDFILE condition
READNEXT command 208
READPREV command 214
SPOOLREAD command 315
WEB READNEXT FORMFIELD command 351
WEB READNEXT HTTPHEADER command 352

ENDFILE option
ISSUE ENDOUTPUT command 157

ENDINPT condition
RECEIVE (non-VTAM) command 239

ENDOUTPUT option
ISSUE ENDFILE command 156

English and katakana characters, mixed 61, 240, 244

ENQ command 92

ENQBUSY condition
ENQ command 93

ENTER option
HANDLE AID command 142

ENTER TRACEID command
monitoring aspects replaced by MONITOR
command 186
tracing aspects replaced by ENTER TRACENUM
command 94

ENTER TRACENUM command 94

ENTRY option
LOAD command 183

entry to assembler-language program 390

ENTRYNAME option
MONITOR command 186

ENVDEFERR condition
RETRIEVE command 251

environment services
commands 10

EOC condition
ALLOCATE (LUTYPE6.1) command 18
CONVERSE (non-VTAM) command 63
CONVERSE (VTAM) command 52
ISSUE RECEIVE command 168
RECEIVE (non-VTAM) command 239
RECEIVE (VTAM) command 229
RECEIVE MAP command 241
RECEIVE PARTN command 245

EODS condition
CONVERSE (VTAM) command 52
ISSUE RECEIVE command 168
RECEIVE (VTAM) command 229
RECEIVE MAP command 241
RECEIVE PARTN command 245

EOF condition
CONVERSE (non-VTAM) command 63
RECEIVE (non-VTAM) command 239

EQUAL option
READ command 198
RESETBR command 247
STARTBR command 327

equated symbols 4

erase all unprotected fields 396

ERASE option
CONVERSE (non-VTAM) command 61
CONVERSE (VTAM) command 50
SEND (non-VTAM) command 280
SEND (VTAM) command 272
SEND CONTROL command 284
SEND MAP command 287
SEND MAP MAPPINGDEV command 291
SEND TEXT command 297
SEND TEXT NOEDIT command 302

ERASEAUP option
SEND CONTROL command 284
SEND MAP command 287
SEND MAP MAPPINGDEV command 291

ERRTERM option
ROUTE command 259

- ESDS (entry-sequenced data set)
 - DELETE command 70
 - READ 199
 - STARTBR command 328
 - WRITE command 359
- ESM
 - ACEE pointer 14
 - QUERY SECURITY command, NOTFND condition 196
 - QUERY SECURITY command, RESCLASS option 195
 - USERNAME 29
- ESM, external security manager 323
- ESMREASON option
 - CHANGE PASSWORD command 37
 - SIGNON command 305
 - VERIFY PASSWORD command 336
- ESMRESP option
 - CHANGE PASSWORD command 37
 - SIGNON command 305
 - VERIFY PASSWORD command 336
- events, timer
 - control area, timer 189
 - monitoring point 186
 - waiting for 339
- EWASUPP option
 - ASSIGN command 25
- examples
 - using the ADDRESS SET command 15
 - using the ASKTIME command 21
 - using the ASSIGN command 29
 - using the BIF DEEDIT command 30
 - using the CANCEL command 13
 - using the DELAY command 67
 - using the DELETE command 72
 - using the DEQ command 78
 - using the DUMP TRANSACTION command 89
 - using the ENQ command 93
 - using the ENTER TRACENUM command 95
 - using the FORMATIME command 112
 - using the FREEMAIN command 117
 - using the GETMAIN command 141
 - using the HANDLE ABEND command 142
 - using the HANDLE AID command 143
 - using the HANDLE CONDITION command 144
 - using the LINK command 181
 - using the LOAD command 185
 - using the MONITOR command 187
 - using the POST command 191
 - using the READ command 203
 - using the READQ TD command 218
 - using the READQ TS command 220
 - using the RELEASE command 246
 - using the RETRIEVE command 251
 - using the REWRITE command 258
 - using the START ATTACH command 324
 - using the START BREXIT command 325
 - using the START command 320

- examples (*continued*)
 - using the WAIT EVENT command 339
 - using the WAIT EXTERNAL command 341
 - using the WAIT JOURNALNAME command 343
 - using the WAITCICS command 346
 - using the WRITE command 363
 - using the WRITE JOURNALNAME command 365
 - using the WRITEQ TD command 370
 - using the XCTL command 375
- EXCEPTION option
 - ENTER TRACENUM command 94
- exception support commands 10
- exclusive control release, UNLOCK command 332
- EXEC CICS command format 1
- EXEC CICS DLI interface
 - No longer documented xv
- execution diagnostic facility (EDF) 323
- exit from ASM program 390
- exit, abnormal termination recovery 141
- expiration time, notification when reached 189
- EXPIRED condition
 - DELAY command 67
 - POST command 190
 - WRITE OPERATOR command 368
- EXPIRYTIME option
 - VERIFY PASSWORD command 336
- EXTATT operand
 - DFHMDI 429
 - DFHMSD 435
- EXTDS option
 - ASSIGN command 25
- external security manager (ESM) 195, 323
- EXTRACT ATTACH (LUTYPE6.1) command 96
- EXTRACT ATTACH (MRO) command 98
- EXTRACT ATTRIBUTES (APPC) command 100
- EXTRACT ATTRIBUTES (MRO) command 101
- EXTRACT CERTIFICATE command 102
- EXTRACT LOGONMSG command 104
- EXTRACT PROCESS command 105
- EXTRACT TCPIP command 107
- EXTRACT TCT command 109
- EXTRACT WEB command 348

F

- FACILITY option
 - ASSIGN command 25
- FCI option
 - ASSIGN command 25, 387
- FCT option
 - DUMP TRANSACTION command 87
- field
 - extracting information 349
- field definition macro, BMS 417
- FIELD option
 - BIF DEEDIT command 30

- field separator operand 430, 436
- FIELD value
 - DFHMDI 429
 - DFHMSD 435
- FIELDS operand
 - DFHMDI 430
- file control
 - commands 10
 - deleting VSAM records 68
 - end browse operation 90
 - read next record 204
 - read previous record 211
 - release exclusive control 332
 - specify start for browse 327
 - update a record 256
 - writing new record 359
- FILE option
 - DELETE command 68
 - ENDBR command 90
 - READ command 198
 - READNEXT command 205
 - READPREV command 212
 - RESETBR command 247
 - REWRITE command 256
 - STARTBR command 327
 - UNLOCK command 332
 - WRITE command 360
- filename
 - definition 2, 3, 4
- filename argument, CICS command format 2
- FILENOTFOUND condition
 - DELETE command 69
 - ENDBR command 90
 - READ command 200
 - READNEXT command 208
 - READPREV command 214
 - RESETBR command 248
 - REWRITE command 257
 - STARTBR command 328
 - UNLOCK command 332
 - WRITE command 361
- FIRST value
 - DFHMDI 430
- FLDSEP operand
 - DFHMDI 430
 - DFHMSD 436
- FLENGTH option
 - DUMP TRANSACTION command 87
 - fullword alternative to LENGTH 393
 - GDS RECEIVE command 131
 - GDS SEND command 133
 - GETMAIN command 140
 - LOAD command 183
 - RECEIVE (non-VTAM) command 238
 - RECEIVE (VTAM) command 228
 - SEND (non-VTAM) command 281
- FLENGTH option (*continued*)
 - SEND (VTAM) command 272
 - SPOOLWRITE command 317
 - WRITE JOURNALNAME command 364
- FMH option
 - CONVERSE (non-VTAM) command 61
 - CONVERSE (VTAM) command 50
 - SEND (non-VTAM) command 281
 - SEND (VTAM) command 272
 - START command 321
- FMHPARM option
 - SEND MAP command 287
 - SEND PAGE command 293
 - SEND TEXT command 297
- FOLD operand
 - DFHMSD 436
- FOR option
 - DELAY command 66
- form field
 - extracting information 349
- FORMATTIME command 110
- FORMFEED option
 - SEND CONTROL command 284
 - SEND MAP command 287
 - SEND MAP MAPPINGDEV command 291
 - SEND TEXT command 297
- FORMFIELD option
 - WEB READ FORMFIELD command 349
 - WEB READNEXT FORMFIELD command 351
 - WEB STARTBROWSE FORMFIELD command 356
- FREE (APPC) command 113
- FREE (LUTYPE6.1) command 114
- FREE (MRO) command 115
- FREE command 112
- free main storage 116
- FREEKB option
 - SEND CONTROL command 284
 - SEND MAP command 287
 - SEND MAP MAPPINGDEV command 292
 - SEND TEXT command 297
 - SEND TEXT NOEDIT command 302
- FREEKB value
 - DFHMDI 429
 - DFHMSD 435
- FREEMAIN command 116
- FROM option
 - CONVERSE (non-VTAM) command 61
 - CONVERSE (VTAM) command 50
 - DUMP TRANSACTION command 87
 - ENTER TRACENUM command 94
 - GDS SEND command 133
 - ISSUE ADD command 148
 - ISSUE PASS command 164
 - ISSUE REPLACE command 170
 - ISSUE SEND command 172
 - RECEIVE MAP command 240

FROM option (*continued*)
 RECEIVE MAP MAPPINGDEV command 242
 REWRITE command 256
 SEND (non-VTAM) command 281
 SEND (VTAM) command 272
 SEND MAP command 287
 SEND MAP MAPPINGDEV command 292
 SEND TEXT command 297
 SEND TEXT MAPPED command 300
 SEND TEXT NOEDIT command 302
 SPOOLWRITE command 317
 START ATTACH command 324
 START command 321
 WRITE command 360
 WRITE JOURNALNAME command 364
 WRITEQ TD command 369
 WRITEQ TS command 371

FROMDOC option
 DOCUMENT INSERT command 82

FROMLENGTH option
 CONVERSE (non-VTAM) command 61
 CONVERSE (VTAM) command 50
 fullword alternative to FROMLENGTH 393

FROMLENGTH option
 CONVERSE (non-VTAM) command 61
 CONVERSE (VTAM) command 50
 ENTER TRACENUM command 94
 fullword length alternative (FROMLENGTH) 393

FRSET option
 SEND CONTROL command 284
 SEND MAP command 287
 SEND MAP MAPPINGDEV command 292

FRSET value
 DFHMDI 429
 DFHMSD 435

FSET value
 DFHMDF 422

Full Function Logical Unit, 3790 48, 227, 270

FULLDATE option
 FORMATTIME 111

fullword length option 393

FUNCERR condition
 ISSUE ABORT command 146
 ISSUE ADD command 148
 ISSUE END command 155
 ISSUE ERASE command 158
 ISSUE NOTE command 163
 ISSUE QUERY command 167
 ISSUE REPLACE command 171
 ISSUE SEND command 173
 ISSUE WAIT command 176

G

GCHARS option
 ASSIGN command 25

GCODES option
 ASSIGN command 25

GDS (generalized data stream) 9
 GDS ALLOCATE command 118
 GDS ASSIGN command 120
 GDS CONNECT PROCESS command 121
 GDS EXTRACT ATTRIBUTES command 123
 GDS EXTRACT PROCESS command 124
 GDS FREE command 125
 GDS ISSUE ABEND command 126
 GDS ISSUE CONFIRMATION command 127
 GDS ISSUE ERROR command 128
 GDS ISSUE PREPARE command 129
 GDS ISSUE SIGNAL command 130
 GDS RECEIVE command 131
 GDS SEND command 133
 GDS WAIT command 135

General Banking Terminal System (2980 General Banking Terminal System) 233

generalized data stream (GDS) 9

generic applid, XRF 23

GENERIC option
 DELETE command 68
 READ command 198
 RESETBR command 247
 STARTBR command 328

GET COUNTER command 136
 GET DCOUNTER command 136

get main storage 139

GETMAIN command 139

GINIT operand
 DFHMDF 422

GMMI option
 ASSIGN command 25

GROUPID option
 SIGNON command 306

GRPNAME operand
 DFHMDF 422

GTEQ option
 READ command 198
 RESETBR command 247
 STARTBR command 328

H

HANDLE ABEND command 141
 HANDLE AID command 142
 HANDLE CONDITION command 143

header
 browsing 346, 347, 356, 357
 extracting information 350, 358
 retrieve next 351, 352

HEADER operand
 DFHMDI 430

HEADER option
 SEND TEXT command 297

- hmmss argument, CICS command format 2
- HIGHLIGHT operand
 - DFHMDF 423
 - DFHMDI 430
 - DFHMSD 436
- HIGHLIGHT option
 - ASSIGN command 26
- HOLD option
 - LOAD command 183
- HONEOM option
 - SEND CONTROL command 284
 - SEND MAP command 288
 - SEND TEXT command 297
 - SEND TEXT NOEDIT command 302
- host command processor LU, 3650/3680 268
- host conversational LU 3650
 - (3270) 46, 267
 - (3653) 46, 268
- HOSTCODEPAGE option
 - WEB READ FORMFIELD command 349
 - WEB RECEIVE command 353
 - WEB STARTBROWSE FORMFIELD command 356
- HOURS option
 - DELAY command 66
 - POST command 190
 - ROUTE command 259
 - START command 321
- HTAB operand
 - DFHMSD 436
- HTTP requests
 - receiving 353, 354
- HTTPHEADER option
 - WEB READ HTTPHEADER command 350
 - WEB READNEXT HTTPHEADER command 352
 - WEB WRITE command 358
- HTTPMETHOD option
 - WEB EXTRACT command 348
- HTTPVERSION option
 - WEB EXTRACT command 348

I

- IC value
 - DFHMDF 422
- IGNORE CONDITION command 144
- IGREQCD condition
 - CONVERSE (VTAM) command 52
 - ISSUE SEND command 173
 - SEND (VTAM) command 273
 - SEND CONTROL command 285
 - SEND MAP command 289
 - SEND PAGE command 294
 - SEND TEXT command 299
 - SEND TEXT MAPPED command 300
 - SEND TEXT NOEDIT command 303
- IGREQID condition
 - ROUTE command 261
 - SEND CONTROL command 285
 - SEND MAP command 289
 - SEND TEXT command 299
 - SEND TEXT MAPPED command 301
 - SEND TEXT NOEDIT command 303
- ILLOGIC condition
 - DELETE command 69
 - EIBRCODE 383
 - ENDBR command 90
 - READ command 200
 - READNEXT command 208
 - READPREV command 214
 - RESETBR command 248
 - REWRITE command 257
 - SPOOLOPEN INPUT command 310
 - SPOOLOPEN OUTPUT command 313
 - SPOOLREAD command 315
 - STARTBR command 328
 - UNLOCK command 333
 - WRITE command 361
- IMMEDIATE option
 - RETURN command 252
- implicit SPOOLCLOSE 310
- INBFMH condition
 - CONVERSE (non-VTAM) command 63
 - CONVERSE (VTAM) command 52
 - RECEIVE (non-VTAM) command 239
 - RECEIVE (VTAM) command 229
- INCREMENT option
 - GET COUNTER command 137
 - GET DCOUNTER command 137
 - REWIND COUNTER command 254
 - REWIND DCOUNTER command 254
- INITIAL operand
 - DFHMDF 423
- initialize main storage 139
- initiate a task 321
- INITIMG option
 - GETMAIN command 140
- INITPARM option
 - ASSIGN command 26
- INITPARMLEN option
 - ASSIGN command 26
- INPARTN option
 - ASSIGN command 26
 - RECEIVE MAP command 240
- input operation without data 397
- INPUTMSG option
 - LINK command 179
 - RETURN command 252
 - XCTL command 374
- INPUTMSGLEN option
 - LINK command 179
 - RETURN command 253

INPUTMSGLEN option (*continued*)
 XCTL command 374
 interactive logical units 47, 226, 269
 interface processor DFHEAI 390
 interface to JES 6
 interpreter logical unit, 3650
 CONVERSE command 45
 ISSUE EODS command 157
 ISSUE LOAD command 162
 RECEIVE command 225
 SEND (VTAM) command 267
 interrogate a data set 167
 interval control
 ASKTIME options 21
 cancel interval control command 35
 CANCEL options 35
 commands 10
 DELAY options 66
 delay processing of task 66
 FORMATTIME options 110
 notification when specified time expires 189
 request current time of day 21
 retrieve data stored for task 249
 start a task 319
 wait for event to occur 339
 INTERVAL option
 DELAY command 66
 POST command 190
 ROUTE command 259
 START command 321
 INTO option
 CONVERSE (non-VTAM) command 61
 CONVERSE (VTAM) command 50
 DOCUMENT RETRIEVE command 84
 EXTRACT LOGONMSG command 104
 GDS RECEIVE command 131
 ISSUE RECEIVE command 168
 READ command 198
 READNEXT command 205
 READPREV command 212
 READQ TD command 217
 READQ TS command 219
 RECEIVE (non-VTAM) command 238
 RECEIVE (VTAM) command 228
 RECEIVE MAP command 240
 RECEIVE MAP MAPPINGDEV command 242
 RECEIVE PARTN command 244
 RETRIEVE command 250
 SPOOLREAD command 315
 WEB RECEIVE command 353
 INVALIDCOUNT option
 VERIFY PASSWORD command 336
 INVERRTERM condition
 ROUTE command 261
 INVITE option
 GDS SEND command 133
 INVITE option (*continued*)
 SEND (non-VTAM) command 281
 SEND (VTAM) command 272
 INVLDC condition
 ROUTE command 261
 SEND CONTROL command 285
 SEND MAP command 290
 SEND TEXT command 299
 INVMP SZ condition
 EIBRCODE byte 3 383
 RECEIVE MAP command 241
 RECEIVE MAP MAPPINGDEV command 243
 SEND MAP command 290
 SEND MAP MAPPINGDEV command 292
 INVOKINGPROG option
 ASSIGN command 26
 INVPARTN condition
 RECEIVE MAP command 241
 RECEIVE PARTN command 245
 SEND CONTROL command 285
 SEND MAP command 290
 SEND TEXT command 299
 SEND TEXT NOEDIT command 303
 INVPARTNSET condition
 SEND PARTNSET command 295
 INVREQ condition
 ALLOCATE (APPC) command 17
 ALLOCATE (LUTYPE6.1) command 18
 ALLOCATE (MRO) command 20
 ASSIGN command 29
 CHANGE PASSWORD command 37
 CHANGE TASK command 38
 CONNECT PROCESS command 40
 CONVERSE (VTAM) command 52
 DELAY command 67
 DELETE command 69
 DELETE COUNTER command 65, 73, 194
 DELETE DOUNTER command 65
 DELETEQ TD command 75
 DELETEQ TS command 76
 DEQ command 77
 DUMP TRANSACTION command 89
 EIBRCODE bytes 1-3 382
 ENDBR command 90
 ENQ command 93
 ENTER TRACENUM command 94
 EXTRACT ATTACH (LUTYPE6.1) command 97
 EXTRACT ATTACH (MRO) command 99
 EXTRACT ATTRIBUTES (APPC) command 100
 EXTRACT CERTIFICATE command 103
 EXTRACT PROCESS command 105
 EXTRACT TCPIP command 108
 EXTRACT TCT command 109
 FORMATTIME command 112
 FREE (APPC) command 113
 FREE (LUTYPE6.1) command 114

INVREQ condition (*continued*)

FREE (MRO) command 115
FREEMAIN command 116
HANDLE AID command 143
ISSUE ABEND command 145
ISSUE ABORT command 146
ISSUE ADD command 149
ISSUE CONFIRMATION command 150
ISSUE END command 155
ISSUE ENDFILE command 156
ISSUE ENDOUTPUT command 157
ISSUE EODS command 157
ISSUE ERASE command 159
ISSUE ERASEAUP command 159
ISSUE ERROR command 161
ISSUE NOTE command 163
ISSUE PASS command 164
ISSUE PREPARE command 165
ISSUE PRINT command 166
ISSUE QUERY command 167
ISSUE RECEIVE command 169
ISSUE REPLACE command 171
ISSUE SEND command 173
ISSUE SIGNAL (APPC) command 174
ISSUE WAIT command 176
LINK command 180
LOAD command 183
MONITOR command 186
POP HANDLE command 188
POST command 190
PURGE MESSAGE command 191
QUERY SECURITY command 196
READ command 201
READNEXT command 208
READPREV command 214
READQ TD command 218
READQ TS command 220
RECEIVE (non-VTAM) command 239
RECEIVE MAP command 241
RECEIVE MAP MAPPINGDEV command 243
RECEIVE PARTN command 245
RELEASE command 245
RESETBR command 248
RETRIEVE command 251
RETURN command 253
REWRITE command 257
ROUTE command 261
SEND (non-VTAM) command 281
SEND CONTROL command 285
SEND MAP command 290
SEND MAP MAPPINGDEV command 292
SEND PAGE command 294
SEND PARTNSET command 295
SEND TEXT command 299
SEND TEXT MAPPED command 301
SEND TEXT NOEDIT command 303

INVREQ condition (*continued*)

SIGNOFF command 304
SIGNON command 306
SPOOLCLOSE command 308
SPOOLOPEN INPUT command 310
SPOOLOPEN OUTPUT command 313
SPOOLREAD command 315
SPOOLWRITE command 317
START ATTACH command 324
START BREXIT command 325
START command 323
STARTBR command 329
SYNCPOINT command 331
SYNCPOINT ROLLBACK command 331
UNLOCK command 333
VERIFY PASSWORD command 337
WAIT CONVID command 337
WAIT EVENT command 339
WAIT EXTERNAL command 341
WAIT TERMINAL command 344
WAITCICS command 346
WEB ENDBROWSE FORMFIELD command 346
WEB ENDBROWSE HTTPHEADER command 347
WEB EXTRACT command 348
WEB READ FORMFIELD command 349
WEB READ HTTPHEADER command 350
WEB READNEXT FORMFIELD command 351
WEB READNEXT HTTPHEADER command 352
WEB RECEIVE command 353
WEB RETRIEVE command 354
WEB SEND command 355
WEB STARTBROWSE FORMFIELD command 356
WEB STARTBROWSE HTTPHEADER command 357
WEB WRITE command 358
WRITE command 361
WRITE JOURNALNAME command 365
WRITE OPERATOR command 368
WRITEQ TD command 369
WRITEQ TS command 372
XCTL command 374

IOERR condition

DELETE command 70
DUMP TRANSACTION command 89
EIBRCODE 383
ENDBR command 90
READ command 201
READNEXT command 208
READPREV command 215
READQ TD command 218
READQ TS command 220
RESETBR command 248
RETRIEVE command 251
REWRITE command 257
START command 323
STARTBR command 329
UNLOCK command 333

IOERR condition (*continued*)
 WRITE command 361
 WRITE JOURNALNAME command 365
 WRITEQ TD command 369
 WRITEQ TS command 372

IOERR option
 WAIT JOURNALNAME command 342

ISCINVREQ condition
 CANCEL command 35
 DELETE command 70
 DELETEQ TD command 75
 DELETEQ TS command 76
 ENDBR command 90
 READ command 201
 READNEXT command 208
 READPREV command 215
 READQ TD command 218
 READQ TS command 220
 RESETBR command 249
 REWRITE command 257
 START command 323
 STARTBR command 329
 UNLOCK command 333
 WRITE command 361
 WRITEQ TD command 369
 WRITEQ TS command 372

ISSUE ABEND command 145
 ISSUE ABORT command 146
 ISSUE ADD command 148
 ISSUE CONFIRMATION command 150
 ISSUE COPY (3270 display) command 151
 ISSUE COPY (3270 logical) command 152
 ISSUE COPY command
 general information 396
 ISSUE DISCONNECT (default) command 153
 ISSUE DISCONNECT (LUTYPE6.1) command 154
 ISSUE DISCONNECT command
 general information 394
 ISSUE END command 155
 ISSUE ENDFILE command 156
 ISSUE ENDOUTPUT command 157
 ISSUE EODS command 157
 ISSUE ERASE command 158
 ISSUE ERASEAUP command 159
 general information 396
 ISSUE ERROR command 161
 ISSUE LOAD command 162
 ISSUE NOTE command 163
 ISSUE PASS command 164
 ISSUE PREPARE command 165
 ISSUE PRINT command 166
 general information 396
 ISSUE QUERY command 167
 ISSUE RECEIVE command 168
 ISSUE REPLACE command 170
 ISSUE RESET command 171
 ISSUE SEND command 172
 ISSUE SIGNAL (APPC) command 174
 ISSUE SIGNAL (LUTYPE6.1) command 175
 ISSUE SIGNAL command
 general information 394
 ISSUE WAIT command 176

ISSUER option
 EXTRACT CERTIFICATE command 102

ITEM option
 READQ TS command 219
 WRITEQ TS command 371

ITEMERR condition
 READQ TS command 220
 WRITEQ TS command 372

IUTYPE option
 BUILD ATTACH (LUTYPE6.1) command 31
 BUILD ATTACH (MRO) command 33
 EXTRACT ATTACH (LUTYPE6.1) command 96
 EXTRACT ATTACH (MRO) command 98

J

JES (job entry subsystem)
 CICS interface to 6
 exits 9
 input 8
 internal limits 6
 output 9
 RESP and RESP2 options 8
 retrieve data from JES spool 7
 send file to destination 8
 spooler commands 8
 write directly to JES spool 7

JIDERR condition
 WRITE JOURNALNAME command 365

JIDERR option
 WAIT JOURNALNAME command 342

JOURNAL command 177

journal control
 create a journal record 177

journal record, creating 364

journaling commands 10

JOURNALNAME option
 WAIT JOURNALNAME command 342
 WRITE JOURNALNAME command 364

JTYPEID option
 WRITE JOURNALNAME command 364

JUSFIRST option
 SEND TEXT command 297

JUSLAST option
 SEND TEXT command 297

JUSTIFY operand
 DFHMDF 423
 DFHMDFI 430

JUSTIFY option
SEND TEXT command 297

K

katakana and English characters, mixed 61, 244

KATAKANA option
ASSIGN command 26

katakana terminals

CONVERSE (3270 logical) command 50
CONVERSE (LUTYPE2/LUTYPE3) command 50
CONVERSE command (3270 display) 61
CONVERSE command (3600 BTAM) 61
CONVERSE command (3735) 61
CONVERSE command (3740) 61
CONVERSE command (System/3) 61
CONVERSE command (System/7) 61
RECEIVE (3270 display) command 238
RECEIVE (3270 logical) command 228
RECEIVE (3790 3270-display) command 238
RECEIVE (LUTYPE2/LUTYPE3) command 228
RECEIVE (System/3) command 238
RECEIVE (System/7) command 238
RECEIVE MAP command 240
RECEIVE PARTN command 244
SEND (3600 BTAM) command 280
SEND (3735) command 280
SEND (3740) command 280
SEND (System/3) command 280
SEND (System/7) command 280

KEEP option
SPOOLCLOSE command 308

KEYLENGTH option
DELETE command 68
ISSUE ERASE command 158
ISSUE REPLACE command 170
READ command 198
READNEXT command 205
READPREV command 212
RESETBR command 247
STARTBR command 328
WRITE command 360

KEYNUMBER option
ISSUE ERASE command 158
ISSUE REPLACE command 170

keyword length 393

L

L40, L64, or L80 options
SEND CONTROL command 284
SEND MAP command 288
SEND TEXT command 298
SEND TEXT NOEDIT command 303

label argument, CICS command format 2

LABEL option
HANDLE ABEND command 142

LANG operand
DFHMSD 436

LANGINUSE option
ASSIGN 26
SIGNON command 306

language codes 409
LANGUAGECODE option
SIGNON command 306

LAST option
GDS SEND command 133
SEND (non-VTAM) command 281
SEND (VTAM) command 272
SEND CONTROL command 284
SEND MAP command 288
SEND PAGE command 293
SEND TEXT command 297
SEND TEXT MAPPED command 300
SEND TEXT NOEDIT command 303

LAST value
DFHMMDI 430

LASTUSETIME option
VERIFY PASSWORD command 336

LDC operand
DFHMSD 436

LDC option
CONVERSE (VTAM) command 50
ROUTE command 259
SEND (VTAM) command 273
SEND CONTROL command 284
SEND MAP command 288
SEND TEXT command 298

LDCMNEM option
ASSIGN command 26

LDCNUM option
ASSIGN command 26

LEAVEKB option
CONVERSE (non-VTAM) command 61
RECEIVE (non-VTAM) command 238
SEND (non-VTAM) command 281

LEFT value
DFHMDF 423
DFHMMDI 430

LENGERR condition
BIF DEEDIT command 30
CONNECT PROCESS command 40
CONVERSE (non-VTAM) command 63
CONVERSE (VTAM) command 52
DEQ command 77
EIBRCODE byte 1 383
ENQ command 93
ENTER TRACENUM command 95
EXTRACT CERTIFICATE command 103
EXTRACT PROCESS command 106
EXTRACT TCPIP command 108

LENGERR condition *(continued)*

- GETMAIN command 141
- ISSUE COPY (3270 logical) command 152
- ISSUE PASS command 164
- ISSUE RECEIVE command 169
- LINK command 180
- LOAD command 184
- QUERY SECURITY command 196
- READ command 201
- READNEXT command 208
- READPREV command 215
- READQ TD command 218
- READQ TS command 220
- RECEIVE (non-VTAM) command 239
- RECEIVE (VTAM) command 229
- RECEIVE PARTN command 245
- RETRIEVE command 251
- RETURN command 253
- REWRITE command 257
- SEND (non-VTAM) command 281
- SEND (VTAM) command 273
- SEND TEXT command 299
- SPOOLOPEN OUTPUT command 313
- SPOOLREAD command 315
- SPOOLWRITE command 317
- START ATTACH command 324
- START BREXIT command 325
- START command 323
- WEB EXTRACT command 348
- WEB READ FORMFIELD command 349
- WEB READ HTTPHEADER command 350
- WEB READNEXT command 352
- WEB READNEXT FORMFIELD command 351
- WEB RECEIVE command 353
- WEB STARTBROWSE FORMFIELD command 356
- WRITE command 361
- WRITE JOURNALNAME command 365
- WRITE OPERATOR command 368
- WRITEQ TD command 369
- WRITEQ TS command 372
- XCTL command 374

LENGERR option

- DOCUMENT RETRIEVE command 84

LENGTH operand

- DFHMDF 423

LENGTH option

- BIF DEEDIT command 30
- built-in function 30
- default (assembler language) 4
- default (C) 3
- default (PL/I) 4
- DEQ command 77
- DOCUMENT RETRIEVE command 84
- DOCUMENT SET command 85
- DUMP TRANSACTION command 87
- ENQ command 92

LENGTH option *(continued)*

- EXTRACT CERTIFICATE command 102
- EXTRACT LOGONMSG command 104
- fullword length alternative (FLENGTH) 393
- GETMAIN command 140
- ISSUE ADD command 148
- ISSUE PASS command 164
- ISSUE RECEIVE command 168
- ISSUE REPLACE command 170
- ISSUE SEND command 172
- LINK command 179
- LOAD command 183
- READ command 198
- READNEXT command 206
- READPREV command 213
- READQ TD command 217
- READQ TS command 219
- RECEIVE (non-VTAM) command 238
- RECEIVE (VTAM) command 228
- RECEIVE MAP command 240
- RECEIVE MAP MAPPINGDEV command 242
- RECEIVE PARTN command 244
- RETRIEVE command 250
- RETURN command 253
- REWRITE command 256
- SEND (non-VTAM) command 281
- SEND (VTAM) command 273
- SEND MAP command 288
- SEND MAP MAPPINGDEV command 292
- SEND TEXT command 298
- SEND TEXT MAPPED command 300
- SEND TEXT NOEDIT command 303
- START ATTACH command 324
- START command 321
- WEB RECEIVE command 353
- WEB SEND command 355
- WRITE command 360
- WRITEQ TD command 369
- WRITEQ TS command 371
- XCTL command 374

LENGTH value

- DFHMDI 428
- DFHMSD 434

LENGTHLIST option

- DUMP TRANSACTION command 88

LIGHTPEN option

- HANDLE AID command 142

LINE operand

- DFHMDI 430

LINE option

- SPOOLWRITE command 317

line value

- DFHMDI 431

line,column value

- DFHMDF 425

- LINEADDR option
 - CONVERSE (non-VTAM) command 61
 - SEND (non-VTAM) command 281
- LINK command 178
- link to program expecting return 178
- LIST option
 - ROUTE command 260
- literal constants 4
- LLID option
 - GDS RECEIVE command 131
- LOAD command 183
- load programs, tables, or maps 183
- LOADING condition
 - DELETE command 70
 - READ command 201
 - READNEXT command 209
 - STARTBR command 329
 - WRITE command 361
- LOCALITY option
 - EXTRACT CERTIFICATE command 102
- LOCALITYLEN option
 - EXTRACT CERTIFICATE command 102
- LOCKED condition
 - DELETE command 70
 - DELETEQ TD command 75
 - DELETEQ TS command 76
 - READ command 202
 - READNEXT command 209
 - READPREV command 215
 - READQ TD command 218
 - REWRITE command 258
 - WRITE command 362
 - WRITEQ TD command 369
 - WRITEQ TS command 372
- logical device code (LDC) 44, 266
- logical messages, BMS
 - completing a logical message 293
 - full BMS
 - ROUTE command 259
 - purging a logical message 191
 - routing a logical message 259
- LOGMESSAGE option
 - QUERY SECURITY command 195
- LOGMODE option
 - ISSUE PASS command 164
- LOGONLOGMODE option
 - ISSUE PASS 164
- LU (logical unit)
 - 3270 Information Display System 44, 152, 223, 265
 - 3270 SCS Printer 43, 264
 - 3270-Display, LUTYPE2 42, 222, 263
 - 3270-Display, LUTYPE3 222, 263
 - 3600 (3601) 44, 224, 266
 - 3600 (3614) 45, 225, 266
 - 3600 pipeline 224, 265
 - 3650 host conversational (3270) 46, 267

- LU (logical unit) *(continued)*
 - 3650 host conversational (3653) 46, 268
 - 3650 interpreter 45, 157, 162, 225, 267
 - 3650/3680 host command processor 268
 - 3770 batch 48, 226, 269
 - 3790 (3270-display) 49, 237, 271
 - 3790 (3270-printer) 271
 - 3790 full-function 227, 270
 - 3790 full-function or inquiry 48
 - 3790 SCS printer 270
 - batch 48, 226, 269
 - conversing with (CONVERSE) 394
 - interactive 47
 - reading data from 168, 393
 - writing data to 148, 393
- LUNAME option
 - ISSUE PASS command 164
- LUTYPE2, 3270-Display LU 42, 222, 263
- LUTYPE3, 3270-Display LU 222, 263
- LUTYPE4
 - logical unit 42, 222, 263
- LUTYPE6.1 logical unit
 - acquiring a session 18
 - communicating on LUTYPE6.1 session 43
 - converting 8-character names to 4 characters 109
 - disconnecting 154
 - ensuring terminal operation has completed 344
 - getting information about 188
 - receiving data 223
 - requesting change of direction 175
 - retrieving values from an LUTYPE6.1 header 96
 - sending data 264
 - specifying values for an MRO attach header 33
 - specifying values for LUTYPE6.1 attach header 31

M

- macros, BMS, summary 417
- magnetic slot reader (MSR) 413
- MAIN option
 - WRITEQ TS command 371
- main storage 139
- map definition macro, BMS 417, 428
- MAP option
 - RECEIVE MAP command 241
 - RECEIVE MAP MAPPINGDEV command 242
 - SEND MAP command 288
 - SEND MAP MAPPINGDEV command 292
- MAP value
 - DFHMSD 438
- MAPATTS operand
 - DFHMDI 431
 - DFHMSD 436
- MAPCOLUMN option
 - ASSIGN command 26

MAPFAIL condition
 RECEIVE MAP command 241
 RECEIVE MAP MAPPINGDEV command 243
 MAPHEIGHT option
 ASSIGN command 26
 MAPLINE option
 ASSIGN command 26
 MAPONLY option
 SEND MAP command 288
 SEND MAP MAPPINGDEV command 292
 MAPONLY value
 DFHMDI 429
 DFHMSD 436
 MAPPINGDEV option
 RECEIVE MAP MAPPINGDEV command 242
 SEND MAP MAPPINGDEV command 292
 maps, loading 183
 mapset definition macro (DFHMSD) 417, 434
 MAPSET option
 RECEIVE MAP command 241
 RECEIVE MAP MAPPINGDEV command 242
 SEND MAP command 288
 SEND MAP MAPPINGDEV command 292
 MAPSFX operand
 DFHPDI 440
 MAPWIDTH option
 ASSIGN command 26
 MASSINSERT option
 WRITE command 360
 MAXLENGTH option
 CONVERSE (non-VTAM) command 61
 CONVERSE (VTAM) command 51
 fullword alternative to MAXLENGTH 393
 GDS RECEIVE command 131
 RECEIVE (non-VTAM) command 238
 RECEIVE (VTAM) command 228
 SPOOLREAD command 315
 MAXIMUM option
 DEFINE COUNTER command 64
 DEFINE DCOUNTER command 64
 QUERY COUNTER command 193
 QUERY DCOUNTER command 193
 MAXLENGTH option
 CONVERSE (non-VTAM) command 61
 CONVERSE (VTAM) command 51
 DOCUMENT RETRIEVE command 84
 fullword length alternative (MAXLENGTH) 393
 RECEIVE (non-VTAM) command 238
 RECEIVE (VTAM) command 228
 WEB RECEIVE command 353
 WRITE OPERATOR command 367
 MAXLIFETIME option
 DEQ command 77
 ENQ command 92
 MAXPROCLen option
 EXTRACT PROCESS command 105
 MAXPROCLen option (*continued*)
 GDS EXTRACT PROCESS command 124
 MCC option
 SPOOLOPEN OUTPUT command 312
 METHODLENGTH option
 WEB EXTRACT command 348
 MINIMUM option
 DEFINE COUNTER command 64
 DEFINE DCOUNTER command 64
 QUERY COUNTER command 193
 QUERY DCOUNTER command 193
 MINUTES option
 DELAY command 66
 POST command 190
 ROUTE command 260
 START command 321
 MMDDYY option
 FORMATTIME command 111
 MMDDYYYY option
 FORMATTIME command 111
 MODE operand
 DFHMSD 436
 model codes (terminal) 387
 MODENAME option
 GDS ALLOCATE command 118
 MONITOR command 186
 monitoring application performance 186
 monitoring commands 10
 MONTHOFYEAR option
 FORMATTIME command 111
 MSR (magnetic slot reader)
 control byte values and constants 413
 DFHMSRCA, 413
 MSR option
 SEND CONTROL command 284
 SEND MAP command 288
 SEND TEXT command 298
 MSRCONTROL option
 ASSIGN command 26
 multi region operation (MRO) commands
 ALLOCATE 20
 BUILD ATTACH 33
 CONVERSE command 53
 EXTRACT ATTACH 98
 EXTRACT ATTRIBUTES 101
 FREE 115
 RECEIVE 230
 SEND 275
 multiple base registers 391
 MUSTENTER value
 DFHMDF 426
 DFHMDI 432
 DFHMSD 438
 MUSTFILL value
 DFHMDF 426
 DFHMDI 432

MUSTFILL value (*continued*)
DFHMSD 438

N

name argument, CICS command format 2

NAME option

WAIT EVENT command 339

WAIT EXTERNAL command 340

WAITCICS command 345

named counter

define named counter 64

delete named counter 73

query named counter 193

named counter server commands 10

named counter server, GET command 136

named counter server, REWIND command 254

named counter server, UPDATE command 334

NAMELENGTH option

WEB READ FORMFIELD command 349

WEB READ HTTPHEADER command 350

WEB READNEXT FORMFIELD command 351

WEB READNEXT HTTPHEADER command 352

WEB STARTBROWSE FORMFIELD command 356

WEB WRITE command 358

naming restriction 389

national language codes 409

NATLANG option

SIGNON command 306

NATLANGINUSE option

ASSIGN command 26

SIGNON command 306

NETNAME option

ASSIGN command 26

EXTRACT TCT command 109

NETNAMEIDERR condition

ALLOCATE (APPC) command 17

new tasks, passing data to 321

NEWPASSWORD option

CHANGE PASSWORD command 37

SIGNON command 306

NEXT option

READQ TS command 219

NEXT value

DFHMDI 428, 430

NEXTTRANSID option

ASSIGN command 27

NLEOM option

ROUTE command 260

SEND MAP command 288

SEND TEXT command 298

NO value

DFHMDI 429, 431

DFHMSD 436, 438

NOAUTOPAGE option

SEND PAGE command 293

NOCC option

SPOOLOPEN OUTPUT command 312

NOCHECK option

START command 322

NODE option

SPOOLOPEN OUTPUT command 312

NODEIDERR condition

SPOOLOPEN OUTPUT command 313

NODUMP option

ABEND command 13

NOFLUSH option

SEND MAP command 289

NOHANDLE option

deactivating HANDLE CONDITION command 144

option 5

overriding HANDLE AID 6

NOJBUFSP condition

WRITE JOURNALNAME command 365

NONVAL condition

ISSUE LOAD command 162

NOPASSBKRD condition

RECEIVE (non-VTAM) command 239

NOPASSBKWR condition

SEND (non-VTAM) command 281

NOQUEUE option

ALLOCATE (APPC) command 16

ALLOCATE (LUTYPE6.1) command 18

ALLOCATE (MRO) command 20

GDS ALLOCATE command 118

NOQUIESCE

ISSUE PASS command 164

NORM value

DFHMDI 422

NOSPACE condition

DUMP TRANSACTION command 89

REWRITE command 258

WRITE command 362

WRITEQ TD command 369

WRITEQ TS command 373

NOSPOOL condition

SPOOLCLOSE command 308

SPOOLOPEN INPUT command 310

SPOOLOPEN OUTPUT command 314

SPOOLREAD command 316

SPOOLWRITE command 317

NOSTART condition

ISSUE LOAD command 162

NOSTG condition

DUMP TRANSACTION command 89

GETMAIN command 141

SPOOLCLOSE command 308

SPOOLOPEN INPUT command 311

SPOOLOPEN OUTPUT command 314

SPOOLREAD command 316

SPOOLWRITE command 317

NOSUSPEND option
 ALLOCATE (APPC) 16
 ALLOCATE (LUTYPE6.1) command 18
 DELETE 68
 ENQ command 93
 GETMAIN command 140
 READ command 199
 READNEXT command 206
 READPREV command 213
 READQ TD command 217
 REWRITE command 256
 WRITE command 360
 WRITE JOURNALNAME command 364
 WRITEQ TS command 372

NOTALLOC condition
 CONNECT PROCESS command 40
 CONVERSE (non-VTAM) command 63
 CONVERSE (VTAM) command 52
 EXTRACT ATTACH (LUTYPE6.1) command 97
 EXTRACT ATTACH (MRO) command 99
 EXTRACT ATTRIBUTES (APPC) command 100
 EXTRACT ATTRIBUTES (MRO) command 101
 EXTRACT LOGONMSG command 104
 EXTRACT PROCESS command 106
 EXTRACT TCT command 109
 FREE (APPC) command 113
 FREE (LUTYPE6.1) command 114
 FREE (MRO) command 115
 FREE command 112
 ISSUE ABEND command 145
 ISSUE CONFIRMATION command 150
 ISSUE COPY (3270 display) command 151
 ISSUE COPY (3270 logical) command 152
 ISSUE DISCONNECT (LUTYPE6.1) command 154
 ISSUE ENDFILE command 156
 ISSUE ENDOUTPUT command 157
 ISSUE EODS command 157
 ISSUE ERASEAUP command 159
 ISSUE ERROR command 161
 ISSUE LOAD command 162
 ISSUE PASS command 164
 ISSUE PREPARE command 165
 ISSUE PRINT command 166
 ISSUE SIGNAL (APPC) command 174
 ISSUE SIGNAL (LUTYPE6.1) command 175
 POINT command 188
 RECEIVE (non-VTAM) command 239
 RECEIVE (VTAM) command 229
 SEND (non-VTAM) command 281
 SEND (VTAM) command 273
 WAIT CONVID command 338
 WAIT SIGNAL command 344
 WAIT TERMINAL command 344

NOTAUTH condition
 CANCEL command 35
 CHANGE PASSWORD command 37

NOTAUTH condition (*continued*)
 DELETE command 71
 DELETEQ TD command 75
 DELETEQ TS command 76
 ENDBR command 90
 HANDLE ABEND command 142
 LINK command 181
 LOAD command 184
 READ command 202
 READNEXT command 209
 READPREV command 215
 READQ TD command 218
 READQ TS command 220
 RELEASE command 246
 RESETBR command 249
 REWRITE command 258
 SIGNON command 306
 SPOOLOPEN INPUT command 311
 START ATTACH command 324
 START BREXIT command 325
 START command 323
 STARTBR command 329
 UNLOCK command 333
 VERIFY PASSWORD command 337
 WRITE command 362
 WRITE JOURNALNAME command 365
 WRITEQ TD command 370
 WRITEQ TS command 373
 XCTL command 374

NOTFND condition
 CANCEL command 36
 DELETE command 71
 DELETE COUNTER command 137, 254, 335
 QUERY SECURITY command 196
 READ command 202
 READNEXT command 209
 READPREV command 215
 RESETBR command 249
 RETRIEVE command 251
 REWRITE 257
 SPOOLCLOSE command 308
 SPOOLOPEN INPUT command 311
 SPOOLOPEN OUTPUT command 314
 SPOOLREAD command 316
 STARTBR command 329
 WEB READ FORMFIELD command 349
 WEB READ HTTPHEADER command 350
 WEB RECEIVE command 353
 WEB SEND command 355
 WEB STARTBROWSE FORMFIELD command 356

NOTFND option
 DOCUMENT RETRIEVE command 84

NOTOPEN condition
 DELETE command 71
 DUMP TRANSACTION command 89
 READ command 202

NOTOPEN condition (*continued*)
 READQ TD command 218
 SPOOLCLOSE command 309
 SPOOLOPEN INPUT command 311
 SPOOLOPEN OUTPUT command 314
 SPOOLREAD command 316
 SPOOLWRITE command 317
 STARTBR command 329
 UNLOCK command 333
 WRITE command 362
 WRITE JOURNALNAME command 365
 WRITEQ TD command 370
 NOTOPEN option
 WAIT JOURNALNAME command 342
 NOTTRUNCATE option
 CONVERSE (non-VTAM) command 62
 CONVERSE (VTAM) command 51
 RECEIVE (non-VTAM) command 238
 RECEIVE (VTAM) command 228
 WEB RECEIVE command 353
 NOWAIT option
 ISSUE ADD command 148
 ISSUE ERASE command 158
 ISSUE REPLACE command 170
 ISSUE SEND command 172
 NUM value
 DFHMDF 422
 number value
 DFHMDF 425
 DFHMDI 428, 430
 NUMEVENTS option
 WAIT EXTERNAL command 340
 WAITCICS command 345
 NUMITEMS option
 READQ TS command 219
 WRITEQ TS command 372
 NUMREC option
 DELETE command 69
 ISSUE ADD command 148
 ISSUE ERASE command 158
 ISSUE REPLACE command 170
 NUMROUTES option
 WRITE OPERATOR command 367
 NUMSEGMENTS option
 DUMP TRANSACTION command 88
 NUMTAB option
 ASSIGN command 27

O
 OBFMT operand
 DFHMDI 431
 DFHMSD 436
 OCCURS operand
 DFHMDF 423

 OFF value
 DFHMDF 423
 DFHMDI 430
 DFHMSD 436
 OIDCARD option
 SIGNON command 306
 OPCLASS option
 ASSIGN command 27
 ROUTE command 260
 OPENERR condition
 DUMP TRANSACTION command 89
 SPOOLOPEN INPUT command 311
 SPOOLOPEN OUTPUT command 314
 OPERID option
 HANDLE AID command 142
 OPERKEYS option
 ASSIGN command 27
 OPERPURGE option
 SEND PAGE command 294
 OPID option
 ASSIGN command 27
 OPSECURITY option
 ASSIGN command 27
 options
 BMS 240, 242, 296
 length 393
 OPTIONS(MAIN)
 in PL/I 390
 ORGABCODE option
 ASSIGN command 27
 ORGANIZATION option
 EXTRACT CERTIFICATE command 102
 ORGANIZATLEN option
 EXTRACT CERTIFICATE command 102
 ORGUNIT option
 EXTRACT CERTIFICATE command 102
 ORGUNITLEN option
 EXTRACT CERTIFICATE command 103
 OUTDESCR option
 SPOOLOPEN OUTPUT command 312
 OUTDESCRERR condition
 SPOOLOPEN OUTPUT command 314
 OUTLINE operand
 DFHMDF 424
 DFHMDI 431
 DFHMSD 436
 OUTLINE option
 ASSIGN command 27
 OUTPARTN option
 SEND CONTROL command 284
 SEND MAP command 289
 SEND TEXT command 298
 SEND TEXT NOEDIT command 303
 output control, 2980 General Banking Terminal System 233
 output to common buffer, 2980 234

OVERFLOW condition
 SEND MAP command 290

OWNER option
 EXTRACT CERTIFICATE command 103

P

PA1-PA3 option
 HANDLE AID command 142

PAGE option
 SPOOLWRITE command 317

PAGENUM option
 ASSIGN command 27

PAGING option
 SEND CONTROL command 285
 SEND MAP command 289
 SEND TEXT command 298
 SEND TEXT MAPPED command 300
 SEND TEXT NOEDIT command 303

partition definition macro (DFHPDI) 418, 440

partition set definition macro (DFHPSD) 418, 441

PARTN operand
 DFHMDI 431
 DFHMSD 436

PARTN option
 RECEIVE PARTN command 244

PARTNER option
 ALLOCATE(APPC) command 16
 CONNECT PROCESS command 39
 GDS ALLOCATE command 118
 GDS CONNECT PROCESS command 121

PARTNERIDERR condition
 ALLOCATE (APPC) command 17
 CONNECT PROCESS command 40

PARTNFAIL condition
 RECEIVE MAP command 241

PARTNPAGE option
 ASSIGN command 27

PARTNS option
 ASSIGN command 27

PARTNSET option
 ASSIGN command 27

PASSBK option
 RECEIVE (non-VTAM) command 239
 SEND (non-VTAM) command 281

passbook control, 2980 233

passing a session 164

passing control
 expecting return (LINK) 178
 on receipt of an AID (HANDLE AID command) 142
 on receipt of an AID (IGNORE AID) 144
 without return (XCTL) 374

passing data to new tasks 321

PASSWORD option
 CHANGE PASSWORD command 37
 SIGNON command 306

PASSWORD option (*continued*)
 VERIFY PASSWORD command 336

PATH option
 WEB EXTRACT command 348

PATHLENGTH option
 WEB EXTRACT command 348

PCT option
 DUMP TRANSACTION command 88

performance, application, monitoring 186

PF1-24 option
 HANDLE AID command 142

PFXLENG option
 WRITE JOURNALNAME command 364

PGMIDERR condition
 HANDLE ABEND command 142
 LINK command 181
 LOAD command 184
 RELEASE command 246
 START BREXIT command 326
 XCTL command 375

PICIN operand
 DFHMDF 424

PICOUT operand
 DFHMDF 424

pipeline logical units 224, 265

PIPLENGTH option
 CONNECT PROCESS command 39
 EXTRACT PROCESS command 105
 GDS CONNECT PROCESS command 121
 GDS EXTRACT PROCESS command 124

PIPLIST option
 CONNECT PROCESS command 39
 EXTRACT PROCESS command 105
 GDS CONNECT PROCESS command 121
 GDS EXTRACT PROCESS command 124

PL/I language
 argument values 3
 LENGTH option default 4
 PROCEDURE statement 390
 STAE option 13
 translated code 389

POINT command 188

POINT option
 MONITOR command 186

pointer-ref argument, CICS command format 2

pointer-value argument, CICS command format 2

POOL option
 DEFINE COUNTER command 64
 DEFINE DCOUNTER command 64
 DELETE COUNTER command 73
 DELETE DCOUNTER command 73
 GET COUNTER command 137
 GET DCOUNTER command 137
 QUERY DCOUNTER command 193
 QUERY COUNTER command 193
 REWIND COUNTER command 254

POOL option (*continued*)
 REWIND DOUNTER command 254
 UPDATE COUNTER command 335
 UPDATE DOUNTER command 335
 POP HANDLE command 188
 PORTNUMBER option
 EXTRACT TCPIP command 107
 PORTNUMNU option
 EXTRACT TCPIP command 107
 POS operand 419
 DFHMDf 425
 POST command 189
 posting timer-event control area 189
 PPT option
 DUMP TRANSACTION command 88
 PREFIX option
 WRITE JOURNALNAME command 365
 PRINCONVID option
 GDS ASSIGN command 120
 PRINSYSID option
 ASSIGN command 27
 GDS ASSIGN command 120
 print displayed information 396
 PRINT option
 ISSUE ABORT command 146
 ISSUE END command 155
 ISSUE SEND command 172
 ISSUE WAIT command 176
 SEND CONTROL command 285
 SEND MAP command 289
 SEND MAP MAPPINGDEV command 292
 SEND TEXT command 298
 SEND TEXT NOEDIT command 303
 SPOOLOPEN OUTPUT command 313
 PRINT value
 DFHMdI 428
 DFHMdS 434
 printer control character list, DFHBMSCA 411
 priority of task, changing 38
 PRIORITY option
 CHANGE TASK command 38
 PROCESS option
 ASSIGN command 27
 BUILD ATTACH (LUTYPE6.1) command 32
 BUILD ATTACH (MRO) command 33
 EXTRACT ATTACH (LUTYPE6.1) command 96
 EXTRACT ATTACH (MRO) command 98
 processing task, control delay of 66
 PROCESSTYPE option
 ASSIGN command 27
 PROCLENGTH option
 CONNECT PROCESS command 39
 EXTRACT PROCESS command 105
 GDS CONNECT PROCESS command 121
 GDS EXTRACT PROCESS command 124
 PROCNAME option
 CONNECT PROCESS command 39
 EXTRACT PROCESS command 105
 GDS CONNECT PROCESS command 121
 GDS EXTRACT PROCESS command 124
 PROFILE option
 ALLOCATE (APPC) command 16
 ALLOCATE (LUTYPE6.1) command 18
 program control
 commands 11
 deleting loaded program 245
 LINK command options 179
 linking to another program 178
 load a program, table, or map 183
 returning program control 252
 transfer program control 374
 PROGRAM option
 ASSIGN command 27
 DUMP TRANSACTION command 88
 HANDLE ABEND command 142
 ISSUE LOAD command 162
 LINK command 179
 LOAD command 183
 RELEASE command 245
 XCTL command 374
 PROT value
 DFHMDf 422
 PROTECT option
 START command 322
 PS operand
 DFHMDf 425
 DFHMdI 431
 DFHMdS 437
 PS option
 ASSIGN command 28
 PSEUDOBIN option
 CONVERSE (non-VTAM) command 62
 RECEIVE (non-VTAM) command 239
 SEND (non-VTAM) command 281
 psid value
 DFHMDf 425
 DFHMdI 431
 DFHMdS 437
 PUNCH option
 SPOOLOPEN OUTPUT command 313
 PURGE MESSAGE command 191
 PURGEABILITY option
 WAIT EXTERNAL command 340
 WAITCICS command 345
 PUSH HANDLE command 192

Q

QBUSY condition
 READQ TD command 218

QIDERR condition
 DELETEQ TD command 75
 DELETEQ TS command 76
 QUERY SECURITY command 196
 READQ TD command 218
 READQ TS command 220
 WRITEQ TD command 370
 WRITEQ TS command 373

QNAME option
 ASSIGN command 28
 DELETEQ TS command 76
 READQ TS command 219
 WRITEQ TS command 372

QUERY COUNTER command 193

QUERY DCOUNTER command 193

QUERY SECURITY command 195

QUERYSTRING option
 WEB EXTRACT command 348

QUERYSTRLEN option
 WEB EXTRACT command 348

QUEUE option
 BUILD ATTACH (LUTYPE6.1) command 32
 BUILD ATTACH (MRO) command 34
 DELETEQ TD command 75
 DELETEQ TS command 76
 EXTRACT ATTACH (LUTYPE6.1) command 97
 EXTRACT ATTACH (MRO) command 99
 READQ TD command 217
 READQ TS command 219
 RETRIEVE command 250
 START command 322
 WRITEQ TD command 369
 WRITEQ TS command 372

QZERO condition
 READQ TD command 218

R

RBA option
 DELETE command 69
 READ command 199
 READNEXT command 206
 READPREV command 213
 RESETBR command 248
 STARTBR command 328
 WRITE command 360

RDATT condition
 CONVERSE (non-VTAM) command 63
 RECEIVE (non-VTAM) command 239
 RECEIVE MAP command 241

reactivate an ABEND exit 141

read attention 56

READ command 197

READ option
 QUERY SECURITY command 195

reading records
 batch data interchange 168
 browsing, next 204
 browsing, previous (VSAM) 211
 file control 197
 from temporary storage queue 219
 from terminal or LU 393
 from transient data queue 217

READNEXT command 204
 READPREV command 211
 READQ TD command 217
 READQ TS command 219
 RECEIVE (2260) command 232
 RECEIVE (2741) command 232
 RECEIVE (2980) command 233
 RECEIVE (3270 display) command 235
 RECEIVE (3270 logical) command 223
 RECEIVE (3600 BTAM) command 235
 RECEIVE (3600 pipeline) command 224
 RECEIVE (3600-3601) command 224
 RECEIVE (3600-3614) command 225
 RECEIVE (3650) command 225
 RECEIVE (3735) command 236
 RECEIVE (3740) command 236
 RECEIVE (3767) command 226
 RECEIVE (3770) command 226
 RECEIVE (3790 3270-display) command 237
 RECEIVE (3790 full-function or inquiry) command 227
 RECEIVE (APPC) command 221
 RECEIVE (LUTYPE2/LUTYPE3) command 222
 RECEIVE (LUTYPE4) command 222
 RECEIVE (LUTYPE6.1) command 223
 RECEIVE (MRO) command 230
 RECEIVE (non-VTAM) command 230
 RECEIVE (System/3) command 231
 RECEIVE (System/7) command 231
 RECEIVE (VTAM default) command 221
 RECEIVE command

input operation without data 397
 read from terminal or logical unit 393
 RECEIVE MAP command 240
 RECEIVE MAP MAPPINGDEV command 242
 RECEIVE PARTN command 244

RECFM option
 BUILD ATTACH (LUTYPE6.1) command 32
 BUILD ATTACH (MRO) command 34
 EXTRACT ATTACH (LUTYPE6.1) command 97
 EXTRACT ATTACH (MRO) command 99

RECORDBUSY condition
 DELETE command 71
 READ command 202
 READNEXT command 209
 READPREV command 215
 REWRITE command 258
 WRITE command 362

RECORDLENGTH option
 SPOOLOPEN OUTPUT command 313

records
 deleting VSAM 68
 reading 168, 197
 release exclusive control 332
 requesting next number 163
 updating 170, 256
 writing new 359
 writing new (adding) 148

REDUCE option
 GET COUNTER command 137
 GET DCOUNTER command 137

register contents in assembler language 390

relative byte address (RBA) 69

RELEASE command 245

RELEASE option
 SEND PAGE command 294

relocatable expression 4

REPEATABLE option
 READ command 199
 READNEXT command 207
 READPREV command 213

REPLY option
 WRITE OPERATOR command 367

REPLYLENGTH option
 WRITE OPERATOR command 368

REQID option
 CANCEL command 35
 DELAY command 66
 ENDBR command 90
 POST command 190
 READNEXT command 207
 READPREV command 213
 RESETBR command 248
 ROUTE command 260
 SEND CONTROL command 285
 SEND MAP command 289
 SEND TEXT command 298
 SEND TEXT MAPPED command 300
 SEND TEXT NOEDIT command 303
 START command 322
 STARTBR command 328
 WAIT JOURNALNAME command 342
 WRITE JOURNALNAME command 365

REQUESTTYPE option
 WEB EXTRACT command 348

RESCLASS option
 QUERY SECURITY command 195

RESET option
 HANDLE ABEND command 142

reset start for browse 247

RESETBR command 247

RESID option
 QUERY SECURITY command 195

RESIDLENGTH option
 QUERY SECURITY command 195

RESOURCE option
 BUILD ATTACH (LUTYPE6.1) command 32
 BUILD ATTACH (MRO) command 34
 DEQ command 77
 ENQ command 93
 ENTER TRACENUM command 94
 EXTRACT ATTACH (LUTYPE6.1) command 97
 EXTRACT ATTACH (MRO) command 99

resource scheduling 77

RESP
 deactivating NOHANDLE 144
 option 5
 values in EIBRESP 383

RESP and RESP2 options
 for interface to JES 8

RESP2
 EXPIRED in messages to console operators 368
 INVREQ in messages to console operators 368
 INVREQ in SIGNOFF command (Security control) 304
 INVREQ in SIGNON (Security control) 306
 INVREQ in WAIT EXTERNAL 341
 INVREQ on WAITCICS 346
 LENGERR in messages to console operators 368
 NOTAUTH in SIGNON (Security control) 306
 option 5
 USERIDERR in SIGNON (Security control) 307
 values in EIBRESP2 384

RESSEC option
 ASSIGN command 28

RESTART option
 ASSIGN command 28

RESTYPE option
 QUERY SECURITY command 195

RETAIN option
 SEND PAGE command 294

RETCODE option
 GDS ALLOCATE command 118
 GDS ASSIGN command 120
 GDS CONNECT PROCESS command 121
 GDS EXTRACT ATTRIBUTES command 123
 GDS EXTRACT PROCESS command 124
 GDS FREE command 125
 GDS ISSUE ABEND command 126
 GDS ISSUE CONFIRMATION command 127
 GDS ISSUE ERROR command 128
 GDS ISSUE PREPARE command 129
 GDS ISSUE SIGNAL command 130
 GDS RECEIVE command 131
 GDS SEND command 133
 GDS WAIT command 135

RETPAGE condition
 SEND CONTROL command 285
 SEND MAP command 290
 SEND PAGE command 294

RETPAGE condition (*continued*)

SEND TEXT command 299

RETRIEVE command 249

retrieve data stored for task 249

RETURN command 252

return program control 252

RETURNPROG option

ASSIGN command 28

REVERSE value

DFHMDf 423

DFHMDI 430

DFHMSD 436

REWIND COUNTER command 254

REWIND DCOUNTER command 254

REWRITE command 256

REWRITE option

WRITEQ TS command 372

RIDFLD option

DELETE command 69

ISSUE ADD command 148

ISSUE ERASE command 158

ISSUE NOTE command 163

ISSUE REPLACE command 170

READ command 199

READNEXT command 207

READPREV command 213

RESETBR command 248

STARTBR command 328

WRITE command 360

RIGHT value

DFHMDf 423

DFHMDI 430

ROLLBACK option

SYNCPOINT ROLLBACK command 331

ROLLEDBACK condition

LINK command 181

SYNCPOINT command 331

ROUTE command 259

ROUTECODES option

WRITE OPERATOR command 368

RPROCESS option

BUILD ATTACH (LUTYPE6.1) command 32

BUILD ATTACH (MRO) command 34

EXTRACT ATTACH (LUTYPE6.1) command 97

EXTRACT ATTACH (MRO) command 99

RRESOURCE option

BUILD ATTACH (LUTYPE6.1) command 32

BUILD ATTACH (MRO) command 34

EXTRACT ATTACH (LUTYPE6.1) command 97

EXTRACT ATTACH (MRO) command 99

RRN option

DELETE command 69

ISSUE ADD command 148

ISSUE ERASE command 158

ISSUE NOTE command 163

ISSUE REPLACE command 170

RRN option (*continued*)

READ command 199

READNEXT command 207

READPREV command 213

RESETBR command 248

STARTBR command 328

WRITE command 360

RTEFAIL condition

ROUTE command 261

RTERMID option

RETRIEVE command 250

START command 322

RTESOME condition

ROUTE command 261

RTRANSID option

RETRIEVE command 250

START command 322

S

SAA (Systems Application Architecture)

communications (CPI) 401

Resource Recovery 399

SADDRLENGTH option

EXTRACT TCPIP command 107

SAME value

DFHMDI 428, 430

schedule use of resource by task 77, 92

scheduling commands 11

SCRNHT option

ASSIGN command 28

SCRNWD option

ASSIGN command 28

SCS (SNA character string)

CONVERSE command 43

SEND (VTAM) command 270

SEND command 264

SCS printer logical unit, 3790 270

SECONDS option

DELAY command 66

POST command 190

ROUTE command 260

START command 322

security commands 11

SEGMENTLIST option

DUMP TRANSACTION command 88

SELNERR condition

ISSUE ABORT command 147

ISSUE ADD command 149

ISSUE END command 156

ISSUE ERASE command 159

ISSUE NOTE command 163

ISSUE QUERY command 167

ISSUE REPLACE command 171

ISSUE SEND command 173

ISSUE WAIT command 177

SEND (2260) command 277
 SEND (2741) command 277
 SEND (2980) command 278
 SEND (3270 display) command 278
 SEND (3270 logical) command 265
 SEND (3600 BTAM) command 279
 SEND (3600 pipeline) command 265
 SEND (3600-3601) command 266
 SEND (3600-3614) command 266
 SEND (3650 interpreter) command 267
 SEND (3650-3270) command 267
 SEND (3650-3653) command 268
 SEND (3650-3680) command 268
 SEND (3735) command 279
 SEND (3740) command 280
 SEND (3767) command 269
 SEND (3770) command 269
 SEND (3790 3270-display) command 271
 SEND (3790 3270-printer) command 271
 SEND (3790 full-function or inquiry) command 270
 SEND (3790 SCS) command 270
 SEND (APPC) command 262
 SEND (LUTYPE2/LUTYPE3) command 263
 SEND (LUTYPE4) command 263
 SEND (LUTYPE6.1) command 264
 SEND (MRO) command 275
 SEND (non-VTAM default) command 275
 SEND (SCS) command 264
 SEND (System/3) command 276
 SEND (System/7) command 276
 SEND (vtam default) command 262
 send asynchronous interrupt 394
 SEND command
 write to terminal 393
 SEND CONTROL command 283
 SEND MAP command 286
 SEND MAP MAPPINGDEV command 291
 SEND PAGE command 293
 SEND PARTNSET command 295
 SEND TEXT command 296
 SEND TEXT MAPPED command 300
 SEND TEXT NOEDIT command 302
 sending data to output device 172
 sequential retrieval, browsing
 reading records 197
 SERIALNUM option
 EXTRACT CERTIFICATE command 103
 SERIALNUMLEN option
 EXTRACT CERTIFICATE command 103
 SERVADDRNU option
 EXTRACT TCPIP command 107
 SERVERADDR option
 EXTRACT TCPIP command 107
 SERVERNAME option
 EXTRACT TCPIP command 107
 SESSBUSY condition
 ALLOCATE (LUTYPE6.1) command 18
 SESSION option
 ALLOCATE (LUTYPE6.1) command 18
 CONNECT PROCESS command 39
 CONVERSE (non-VTAM) command 62
 CONVERSE (VTAM) command 51
 EXTRACT ATTACH (LUTYPE6.1) command 97
 EXTRACT ATTACH (MRO) command 99
 EXTRACT ATTRIBUTES (MRO) command 101
 FREE (LUTYPE6.1) command 114
 FREE (MRO) command 115
 ISSUE DISCONNECT (LUTYPE6.1) command 154
 ISSUE SIGNAL (LUTYPE6.1) command 175
 POINT command 188
 RECEIVE (non-VTAM) command 239
 RECEIVE (VTAM) command 228
 SEND (non-VTAM) command 281
 SEND (VTAM) command 273
 WAIT TERMINAL command 344
 session, passing 164
 SESSIONERR condition
 ALLOCATE (LUTYPE6.1) command 18
 EIBRCODE bytes 1-2 382
 SET option
 ADDRESS SET command 15
 CONVERSE (non-VTAM) command 62
 CONVERSE (VTAM) command 51
 EXTRACT LOGONMSG command 104
 GDS RECEIVE command 131
 GETMAIN command 140
 ISSUE RECEIVE command 168
 LOAD command 183
 POST command 190
 READ command 199
 READNEXT command 207
 READPREV command 213
 READQ TD command 217
 READQ TS command 219
 RECEIVE (non-VTAM) command 239
 RECEIVE (VTAM) command 228
 RECEIVE MAP command 241
 RECEIVE MAP MAPPINGDEV command 243
 RECEIVE PARTN command 244
 RETRIEVE command 250
 SEND CONTROL command 285
 SEND MAP command 289
 SEND MAP MAPPINGDEV command 292
 SEND PAGE command 294
 SEND TEXT command 298
 WEB READ FORMFIELD command 349
 WEB RECEIVE command 353
 SHARED option
 GETMAIN command 140
 SIGDATA option
 ASSIGN command 28

SIGNAL condition
 CONVERSE (VTAM) command 52
 ISSUE CONFIRMATION command 150
 ISSUE DISCONNECT (default) command 153
 ISSUE ERROR command 161
 RECEIVE (VTAM) command 229
 SEND (VTAM) command 273
 WAIT SIGNAL command 344
 WAIT TERMINAL command 344

SIGNOFF command 304

SIGNON command 305

single thread used with JES 310

SIT option
 DUMP TRANSACTION command 88

SIZE operand
 DFHMDI 431

SNAMELENGTH option
 EXTRACT TCPIP command 107

SOSI operand
 DFHMDF 426
 DFHMDI 431
 DFHMSD 437

SOSI option
 ASSIGN command 28

SPCOMMAND
 RESID value not valid 196
 RESTYPE values 196

SPOLBUSY condition
 SPOOLOPEN INPUT command 311
 SPOOLOPEN OUTPUT command 314

SPOLERR condition
 SPOOLOPEN INPUT command 311
 SPOOLREAD command 316
 SPOOLWRITE command 318

Spool commands 11

SPOOLCLOSE command 308

SPOOLCLOSE, implicit 310

spooler, JES 6

SPOOLOPEN INPUT command 310

SPOOLOPEN OUTPUT 312

SPOOLREAD command 315

SPOOLWRITE command 317

SSLTYPE option
 EXTRACT TCPIP command 108

STAE option, PL/I 13

standard attribute and printer control character list, BMS (DFHBMSCA) 411

START ATTACH command 324

START command 319, 325

STARTBR command 327

STARTCODE option
 ASSIGN command 28

STATE option
 ALLOCATE (APPC) command 16
 ALLOCATE (MRO) command 20
 CONNECT PROCESS command 39

STATE option (*continued*)
 CONVERSE (non-VTAM) command 62
 CONVERSE (VTAM) command 51
 EXTRACT ATTRIBUTES (APPC) command 100
 EXTRACT ATTRIBUTES (MRO) command 101
 EXTRACT CERTIFICATE command 103
 FREE (APPC) command 113
 FREE (MRO) command 115
 GDS ALLOCATE command 118
 GDS CONNECT PROCESS command 121
 GDS EXTRACT ATTRIBUTES command 123
 GDS FREE command 125
 GDS ISSUE ABEND command 126
 GDS ISSUE CONFIRMATION command 127
 GDS ISSUE ERROR command 128
 GDS ISSUE PREPARE command 129
 GDS ISSUE SIGNAL command 130
 GDS RECEIVE command 132
 GDS SEND command 133
 GDS WAIT command 135
 ISSUE ABEND command 145
 ISSUE CONFIRMATION command 150
 ISSUE ERROR command 161
 ISSUE PREPARE command 165
 ISSUE SIGNAL (APPC) command 174
 RECEIVE (non-VTAM) command 239
 RECEIVE (VTAM) command 229
 SEND (non-VTAM) command 281
 SEND (VTAM) command 273
 WAIT CONVID command 337

STATELEN option
 EXTRACT CERTIFICATE command 103

STATIONID option
 ASSIGN command 28

STATUSCODE option
 WEB SEND command 355

STATUSTEXT option
 WEB SEND command 355

storage area length 23

storage control commands 11

STORAGE operand
 DFHMSD 437

STORAGE option
 DUMP TRANSACTION command 88

storage, dynamic 391

STRELERR condition
 SPOOLCLOSE command 309
 SPOOLOPEN INPUT command 311
 SPOOLOPEN OUTPUT command 314
 SPOOLREAD command 316
 SPOOLWRITE command 318

STRFIELD option
 CONVERSE (non-VTAM) command 62
 CONVERSE (VTAM) command 51
 SEND (non-VTAM) command 281
 SEND (VTAM) command 273

- stub, program 390
- SUBADDR option
 - ISSUE ABORT command 146
 - ISSUE END command 155
 - ISSUE SEND command 172
 - ISSUE WAIT command 176
- SUFFIX operand
 - DFHMSD 437
 - DFHPSD 441
- SUPPRESSED condition
 - DUMP TRANSACTION command 89
 - GET COUNTER command 138
 - REWIND COUNTER command 255
 - UPDATE COUNTER command 335
 - WRITE command 362
- SUSPEND command 330
- switched line disconnection 394
- SYMBOL option
 - DOCUMENT INSERT command 83
 - DOCUMENT SET command 85
- SYMBOLERR condition
 - DOCUMENT SET command 86
- symbolic register DFHEIPLR 392
- SYMBOLLIST option
 - DOCUMENT CREATE command 80
 - DOCUMENT SET command 85
- synchronization levels
 - basic conversations 124
- synchronize, action
 - journal output (WAIT JOURNALNAME) 342
 - terminal input/output 393
- SYNCLEVEL option
 - CONNECT PROCESS command 39
 - EXTRACT PROCESS command 105
 - GDS CONNECT PROCESS command 122
 - GDS EXTRACT PROCESS command 124
- SYNCONRETURN option
 - LINK command 180
- syncpoint
 - backing out 331
 - commands 11
 - establishing 331
- SYNCPOINT command 331
- SYNCPOINT ROLLBACK command 331
- syntax notation 1
- SYSBUSY condition
 - ALLOCATE (APPC) command 17
 - ALLOCATE (LUTYPE6.1) command 18
 - ALLOCATE (MRO) command 20
 - EIBRCODE byte 3 382
- SYSID option
 - ALLOCATE (APPC) command 16
 - ALLOCATE (LUTYPE6.1) command 18
 - ALLOCATE (MRO) command 20
 - ASSIGN command 28
 - CANCEL command 35
- SYSID option (*continued*)
 - DELETE command 69
 - DELETEDQ TD command 75
 - DELETEDQ TS command 76, 372
 - ENDBR command 90
 - EXTRACT TCT command 109
 - GDS ALLOCATE command 118
 - LINK command 180
 - READ command 200
 - READNEXT command 207
 - READPREV command 214
 - READQ TD command 218
 - READQ TS command 220
 - RESETBR command 248
 - REWRITE command 256
 - START command 322
 - STARTBR command 328
 - UNLOCK command 332
 - WRITE command 360
 - WRITEQ TD command 369
- SYSIDERR condition
 - ALLOCATE (APPC) command 17
 - ALLOCATE (LUTYPE6.1) command 19
 - ALLOCATE (MRO) command 20
 - CANCEL command 36
 - DELETE command 71
 - DELETEDQ TD command 75
 - DELETEDQ TS command 76
 - EIBRCODE bytes 1-2 382
 - ENDBR command 91
 - LINK command 181
 - READ command 202
 - READNEXT command 209
 - READPREV command 215
 - READQ TD command 218
 - READQ TS command 220
 - RESETBR command 249
 - REWRITE command 258
 - START command 323
 - STARTBR command 330
 - UNLOCK command 333
 - WRITE command 362
 - WRITEQ TD command 370
 - WRITEQ TS command 373
- System/3
 - CONVERSE command 54
 - RECEIVE command 231
 - SEND command 276
- System/7
 - CONVERSE command 54
 - RECEIVE command 231
 - SEND command 276
- systemname
 - definition 2, 3, 4
- systemname argument, CICS command format 2

T

- TABLES option
 - DUMP TRANSACTION command 88
- tables, loading 183
- task
 - initiation 321
- task control commands 11
- TASK option
 - DUMP TRANSACTION command 88
- task, abnormal termination 141
- task, delay processing of 66
- TASKDATALOC resource definition option 14
- TASKPRIORITY option
 - ASSIGN command 28
- TCAM-supported terminals and logical units 394
- TCP/IP services 11
- TCPIPSERVICE option
 - EXTRACT TCPIP command 108
- TCT option
 - DUMP TRANSACTION command 88
- TCTUA option
 - ADDRESS command 15
- TCTUALENG option
 - ASSIGN command 29
- teletypewriter
 - messages 395
 - programming 395
- TELLERID option
 - ASSIGN command 29
- TEMPLATE option
 - DOCUMENT INSERT command 83
- temporary storage control commands 11
- TERM operand
 - DFHMDI 431
 - DFHMSD 437
- TERMCODE option
 - ASSIGN command 29, 387
- TERMERR condition
 - CONNECT PROCESS command 40
 - CONVERSE (non-VTAM) command 63
 - CONVERSE (VTAM) command 52
 - ISSUE ABEND command 145
 - ISSUE CONFIRMATION command 150
 - ISSUE COPY (3270 logical) command 152
 - ISSUE DISCONNECT (default) command 153
 - ISSUE DISCONNECT (LUTYPE6.1) command 154
 - ISSUE EODS command 157
 - ISSUE ERASEAUP command 160
 - ISSUE ERROR command 161
 - ISSUE LOAD command 162
 - ISSUE PREPARE command 165
 - ISSUE PRINT command 166
 - ISSUE SIGNAL (APPC) command 174
 - ISSUE SIGNAL (LUTYPE6.1) command 175
 - LINK command 181
- TERMERR condition (*continued*)
 - RECEIVE (non-VTAM) command 239
 - RECEIVE (VTAM) command 229
 - SEND (non-VTAM) command 281
 - SEND (VTAM) command 274
 - WAIT SIGNAL command 344
- TERMINAL option
 - EXTRACT TCT command 109
 - ISSUE COPY (3270 display) command 151
 - ISSUE COPY (3270 logical) command 152
 - START command 322
- TERMINID option
 - EXTRACT TCT command 109
 - ISSUE COPY (3270 display) command 151
 - ISSUE COPY (3270 logical) command 152
 - START command 322
- TERMIDERR condition
 - ISSUE COPY (3270 display) command 151
 - START command 323
- terminal control 393
 - commands 11
- terminal model codes 387
- terminal operator paging, initiate paging transaction 293
- TERMINAL option
 - DUMP TRANSACTION command 88
 - RECEIVE MAP command 241
 - SEND CONTROL command 285
 - SEND MAP command 289
 - SEND TEXT command 299
 - SEND TEXT MAPPED command 300
 - SEND TEXT NOEDIT command 303
- terminal type codes 387
- terminate data set processing
 - abnormal 146
 - normal 155
- TERMPRIORITY option
 - ASSIGN command 29
- TEXT option
 - DOCUMENT INSERT command 83
 - WRITE OPERATOR command 368
- TEXTKYBD option
 - ASSIGN command 29
- TEXTLENGTH option
 - WRITE OPERATOR command 368
- TEXTPRINT option
 - ASSIGN command 29
- time of day, requesting 21
- TIME option
 - DELAY command 66
 - FORMATTIME command 111
 - POST command 190
 - ROUTE command 260
 - START command 322
- TIMEOUT option
 - WRITE OPERATOR command 368
- timer-event control area 189
- TIMESEP option
 - FORMATTIME command 111
- TIOAPFX operand
 - DFHMDI 431
 - DFHMSD 438

TITLE option
 ROUTE command 260

TO option
 DOCUMENT INSERT command 83

TOLENGTH option
 CONVERSE (non-VTAM) command 62
 CONVERSE (VTAM) command 51
 fullword alternative to TOLENGTH 393
 SPOOLREAD command 315

TOKEN option
 DELETE command 69
 READ command 200
 READNEXT 207
 READPREV command 214
 REWRITE command 256
 SPOOLCLOSE command 308
 SPOOLOPEN INPUT command 310
 SPOOLOPEN OUTPUT command 313
 SPOOLREAD command 315
 SPOOLWRITE command 317
 UNLOCK command 332

TOLENGTH option
 CONVERSE (non-VTAM) command 62
 CONVERSE (VTAM) command 51
 fullword length alternative (TOLENGTH) 393

TRACENUM option
 ENTER TRACENUM command 94

TRAILER operand
 DFHMDI 431

TRAILER option
 SEND PAGE command 294
 SEND TEXT command 299

TRANPRIORITY option
 ASSIGN command 29

transfer program control 374

TRANSID option
 CANCEL command 35
 LINK command 180
 RETURN command 253
 SEND PAGE command 294
 START ATTACH command 324
 START BREXIT command 325
 START command 322

TRANSIDERR condition
 START ATTACH command 324
 START BREXIT command 326
 START command 323

transient data commands 12

transient data control
 delete intrapartition queue 75
 read data from TD queue 217
 write data to TD queue 369

translated code 389

TRANSP operand
 DFHMDF 426
 DFHMDI 432

TRANSP operand (*continued*)
 DFHMSD 438

TRIGGER option
 HANDLE AID command 142

TRIGGER value
 DFHMDF 426
 DFHMDI 432
 DFHMSD 438

TRIGRAPH operand
 DFHMSD 438

TRT option
 DUMP TRANSACTION command 89

TSIOERR condition
 PURGE MESSAGE command 191
 SEND CONTROL command 285
 SEND MAP command 290
 SEND PAGE command 294
 SEND TEXT command 299
 SEND TEXT MAPPED command 301
 SEND TEXT NOEDIT command 304

TWA option
 ADDRESS command 15

TWALENG option
 ASSIGN command 29

type codes (terminal) 387

TYPE operand
 DFHMSD 438

TYPE option
 WEB RECEIVE command 353

U

UNATTEND option
 ASSIGN command 29

UNCOMMITTED
 READ command 200

UNCOMMITTED option
 READNEXT 207
 READPREV command 214

UNDERLINE value
 DFHMDF 423
 DFHMDI 430
 DFHMSD 436

UNEXPIN condition
 ISSUE ABORT command 147
 ISSUE ADD command 149
 ISSUE END command 156
 ISSUE ERASE command 159
 ISSUE NOTE command 163
 ISSUE QUERY command 167
 ISSUE RECEIVE command 169
 ISSUE REPLACE command 171
 ISSUE SEND command 173
 ISSUE WAIT command 177
 RECEIVE MAP command 241

- UNLOCK command 332
- UNPROT value
 - DFHMDF 422
- UNTIL option
 - DELAY command 67
- UPDATE COUNTER command 334
- UPDATE DCOUNTER command 334
- UPDATE option
 - QUERY SECURITY command 196
 - READ command 200
 - READNEXT 207
 - READPREV command 214
- updating records
 - batch data interchange 170
 - file control 256
- USERDATAKEY option
 - GETMAIN command 141
- USEREXIT value
 - DFHMDF 426
 - DFHMDI 432
 - DFHMSD 438
- USERID option
 - ASSIGN command 29
 - CHANGE PASSWORD command 37
 - EXTRACT CERTIFICATE command 103
 - SIGNON command 306
 - SPOOLOPEN INPUT command 310
 - SPOOLOPEN OUTPUT command 313
 - START BREXIT command 325
 - START command 323
 - VERIFY PASSWORD command 336
- USERIDERR condition
 - CHANGE PASSWORD command 37
 - SIGNON command 307
 - START BREXIT command 326
 - START command 323
 - VERIFY PASSWORD command 337
- USERNAME option
 - ASSIGN command 29
- USERPRIORITY option
 - ASSIGN command 29
- USING option
 - ADDRESS SET command 15

V

- VALIDATION option
 - ASSIGN command 29
- VALIDN operand
 - DFHMDF 426
 - DFHMDI 432
 - DFHMSD 438
- VALUE option
 - DEFINE COUNTER command 65
 - DEFINE DCOUNTER command 65
 - DOCUMENT SET command 86

- VALUE option (*continued*)
 - GET COUNTER command 137
 - GET DCOUNTER command 137
 - QUERY COUNTER command 193
 - QUERY DCOUNTER command 193
 - UPDATE COUNTER command 335
 - UPDATE DCOUNTER command 335
 - WEB READ FORMFIELD command 349
 - WEB READ HTTPHEADER command 350
 - WEB READNEXT FORMFIELD command 351
 - WEB READNEXT HTTPHEADER command 352
 - WEB WRITE command 358
- VALUELENGTH option
 - WEB READ FORMFIELD command 349
 - WEB READ HTTPHEADER command 350
 - WEB READNEXT FORMFIELD command 351
 - WEB READNEXT HTTPHEADER command 352
 - WEB WRITE command 358
- VERIFY PASSWORD command 336
- VIEWPOS operand
 - DFHPDI 440
- VIEWSIZE operand
 - DFHPDI 440
- VOLUME option
 - ISSUE ABORT command 146
 - ISSUE ADD command 148
 - ISSUE END command 155
 - ISSUE ERASE command 158
 - ISSUE NOTE command 163
 - ISSUE QUERY command 167
 - ISSUE REPLACE command 170
 - ISSUE SEND command 172
 - ISSUE WAIT command 176
- VOLUMELENG option
 - ISSUE ABORT command 146
 - ISSUE ADD command 148
 - ISSUE END command 155
 - ISSUE ERASE command 158
 - ISSUE NOTE command 163
 - ISSUE QUERY command 167
 - ISSUE REPLACE command 170
 - ISSUE SEND command 172
 - ISSUE WAIT command 176
- VSAM WRITE MASSINSERT
 - DISABLED cannot occur 332
 - NOTOPEN cannot occur 333
 - terminate operation 332
- VTAB operand
 - DFHMSD 439
- VTAM logon data, access to 104

W

- WAIT CONVID (APPC) command 337
- WAIT EVENT command 339

- WAIT EXTERNAL command 340
- WAIT JOURNALNAME command 342
- WAIT JOURNALNUM command 343
- WAIT option
 - GDS SEND command 134
 - ISSUE COPY (3270 display) command 151
 - ISSUE COPY (3270 logical) command 152
 - ISSUE ERASEAUP command 159
 - RETRIEVE command 250
 - SEND (non-VTAM) command 281
 - SEND (VTAM) command 273
 - SEND command 393
 - SEND CONTROL command 285
 - SEND MAP command 289
 - SEND TEXT command 299
 - SEND TEXT MAPPED command 300
 - SEND TEXT NOEDIT command 303
 - terminal control 393
 - WRITE JOURNALNAME command 365
- WAIT SIGNAL command 344
- WAIT TERMINAL command 344
 - general information 393
- WAITCICS command 345
- waits
 - batch data interchange 176
 - for event to occur 339
 - terminal control operation 393
- WEB ENDBROWSE FORMFIELD command 346
- WEB ENDBROWSE HTTPHEADER command 347
- WEB EXTRACT command 348
- WEB READ FORMFIELD command 349
- WEB READ HTTPHEADER command 350
- WEB READNEXT FORMFIELD command 351
- WEB READNEXT HTTPHEADER command 352
- WEB RECEIVE command 353
- WEB RETRIEVE command 354
- WEB SEND command 355
- web services 12
- WEB STARTBROWSE FORMFIELD command 356
- WEB STARTBROWSE HTTPHEADER command 357
- WEB WRITE command 358
- WPMEDIA option
 - ISSUE ABORT command 146
 - ISSUE END command 155
 - ISSUE SEND command 173
 - ISSUE WAIT command 176
- WRAP option
 - GET COUNTER command 137
 - GET DCOUNTER command 137
- WRBRK condition
 - CONVERSE (non-VTAM) command 63
 - SEND (non-VTAM) command 282
 - SEND CONTROL command 285
 - SEND MAP command 290
 - SEND PAGE command 294
 - SEND TEXT command 299

- WRBRK condition (*continued*)
 - SEND TEXT MAPPED command 301
 - SEND TEXT NOEDIT command 304
- write break 56
- WRITE command 359
- WRITE JOURNALNAME command 364
- WRITE JOURNALNUM command 366
- WRITE OPERATOR command 367
 - critical action 367
 - eventual action 367
 - immediate action 367
- WRITEQ TD command 369
- WRITEQ TS command 371
- writing data
 - to temporary storage queue 371
 - to terminal or logical unit 393
 - to transient data queue 369
- writing records to data sets
 - batch data interchange 148
 - file control 359

X

- XCTL command 374
- XINIT operand
 - DFHMDF 426
- XRF, generic applid 23

Y

- YEAR option
 - FORMATTIME command 111
- YES value
 - DFHMDF 429, 431
 - DFHMDF 436, 438
- YYDDD option
 - FORMATTIME command 111
- YYDDMM option
 - FORMATTIME command 111
- YYMMDD option
 - FORMATTIME command 111
- YYYYDDD option
 - FORMATTIME command 111
- YYYYDDMM option
 - FORMATTIME command 111
- YYYYMMDD option
 - FORMATTIME command 111

Z

- ZERO value
 - DFHMDF 423

Sending your comments to IBM

If you especially like or dislike anything about this book, please use one of the methods listed below to send your comments to IBM.

Feel free to comment on what you regard as specific errors or omissions, and on the accuracy, organization, subject matter, or completeness of this book.

Please limit your comments to the information in this book and the way in which the information is presented.

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, to this address:
 - Information Development Department (MP095)
 - 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 44–1962–870229
 - From within the U.K., use 01962–870229
- Electronically, use the appropriate network ID:
 - IBM Mail Exchange: GBIBM2Q9 at IBMMAIL
 - IBMLink™: HURSLEY(IDRCF)
 - Internet: idrcf@hursley.ibm.com

Whichever you use, ensure that you include:

- The publication number and title
- The topic to which your comment applies
- Your name and address/telephone number/fax number/network ID.



Program Number: 5655-147



Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.

SC33-1688-34





CICS TS for OS/390

CICS Application Programming Reference

Release 3